βetaWorks

IBM Integration Bus

Workload Management
Unresponsive Flows

June, 2013

Hands-on lab built at product
code level Version 9.0.0.0

1. Introduction to Unresponsive Flows

When a message flow has an undetected design flaw, it can become unresponsive. This may be due an excessive "sleep" time in an ESQL command, a poison message causing a backout, or another looping condition. Prior to IIB V9, an operator needed to stop the entire execution group to interrupt the unresponsive flow. This impacts all other flows running in the execution group.

In IIB V9, new function has been added to flush a runaway message flow from the system. There are three methods which can be used to remove the message flow.

1. Programmatically check from within a message flow to see if it has been requested to stop
2. Manually force a message flow to stop
3. Automatically force a message flow to stop

In IIB V9 a message flow developer can use programming APIs to check to see if a request has been made to stop processing within a message flow. This will be discussed in the next section but will not be exercised in this lab. You can briefly read about its capabilities. You can specify and monitor the maximum amount of time that any message flow is allowed to process a message for, and to specify an action to be taken if the time-out value is exceeded. Additionally, manual requests can be made to stop a message flow by restarting the execution group. We will exercise both of these methods in this lab.

2. Programmatically check from within a message flow if it has been requested to stop

In this section we will briefly review the three programming APIs used to stop and remove unresponsive flows. As noted earlier, this section is just for review. Only methods two and three will be exercised in the lab.

The three programming APIs (ESQL, Java™, and .NET) have new functions added to them that allows you to check from within a message flow if the flow has been requested to stop. The function in all three APIs returns a Boolean value of true if a request has been made to stop the message flow.

The functions are described as follows:

ESQL

```
INSTANCESTOPPING();
```

Returns true if a request has been made to stop the message flow. For more information, see INSTANCESTOPPING function in the IIB V9 Infocenter.

Java

```
Static Boolean MbMessageFlow.isInstanceStopping();
```

Returns true if a request has been made to stop the message flow. For more information, see MbMessageFlow Class under Java user-defined extensions API in the IIB V9 Infocenter.

.NET

```
Static Boolean NbMessageFlow.InstanceStopping();
```

Returns true if a request has been made to stop the message flow. For more information, see NbMessageFlow Class under .NET reference in the IIB V9 Infocenter.

3. Manually force a message flow to stop

The mqsistopmsgflow command has been enhanced with a force option to escalate the mechanisms that are used to stop a message flow. A force option –f has been added to the mqsistopmsgflow command to allow you to specify how the message flow can be forced to stop.

Without the force option, the mqsistopmsgflow command sends a request to a message flow to stop it by first waiting for all threads that are used by the message flow to finish. If one of the threads is stuck in an operation, then the message flow stop will never complete.

The force option -f has been added to allow system administrators to specify how the message flow should be stopped. Currently, *restartExecutionGroup* is the only valid option available, that causes the message flow to be flagged as stopped and then to restart the execution group. When the execution group restarts the message flow will be in the stop state.

You can combine using the mqsistopmsgflow command without, and then with, the force option to escalate the mechanisms that are used to stop the message flow.

For example:

```
mqsistopmsgflow IB9NODE –e eg1 –m mf1 –w 30
mqsistopmsgflow IB9NODE –e eg1 –m mf1 –w 30 –f restartExecutionGroup
```

The first command would first try to the stop the message flow normally and wait 30 seconds for a response. The second command would cause the execution group to restart, but only if the flow was not already stopped.

The non-force version of the mqsistopmsgflow command has the potential to hang the entire execution group deployment mechanism, but the force version of the mqsistopmsgflow command will not suffer from the same issue and avoids taking any locks that might cause hanging or deadlocks. The force version of the mqsistopmsgflow command will still work even if the standard deployment mechanisms are hung.

The same force option is available in the CMP and REST APIs, and within the web user interface.

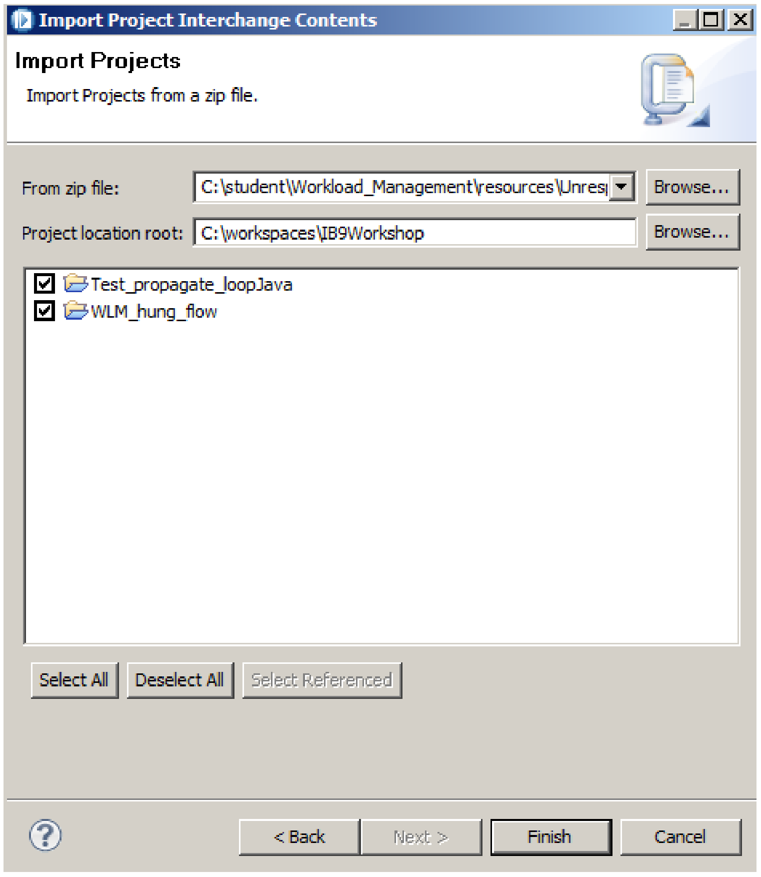
3.1 Import the Application

1. In the Integration Toolkit, import a Project Interchange file.

   Navigate to ***c:\student\Workload_Management\resources\UnrespFlows***:

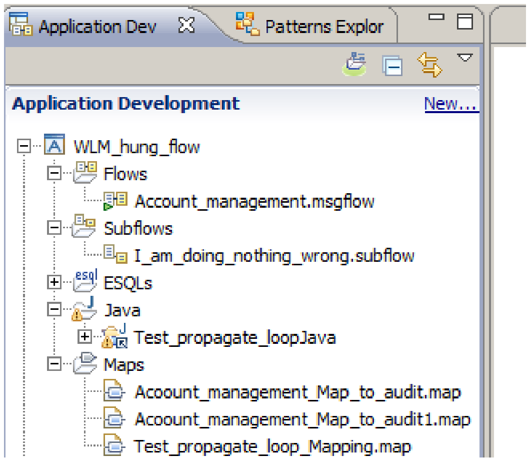
   Select WLM_UnrespFlows.zip and then Open.

2. Make sure all folders are checked and click Finish.


Import Project Interchange Contents
Import Projects
Import Projects from a zip file.
From zip file:
C:\student\Workload_Management\resources\Unres|
Browse...
Project location root:
C:\workspaces\IB9Workshop
Browse...
☑ Test_propagate_loopJava
☑ WLM_hung_flow
Select All
Deselect All
Select Referenced
< Back
Next >
Finish
Cancel

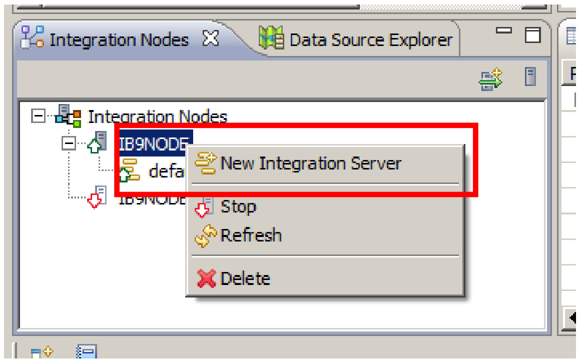
3. You now have an application named WLM_hung_flow. Expand WLM_hung_flow to see the sub folders Flows, Subflows, ESQLs, Java, and Maps.

   Review the Account_management.msgflow. This is the flow that we will hang and attempt to force it out of the execution group.

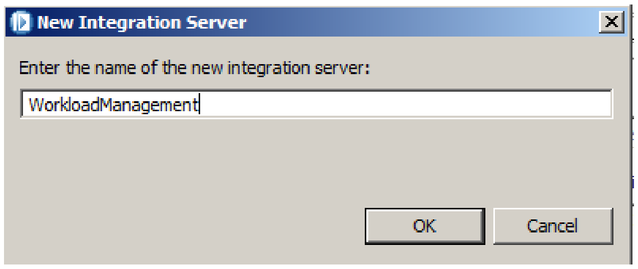

Application Dev
Patterns Explor
Application Development
New...
WLM_hung_flow
Flows
Account_management.msgflow
Subflows
I_am_doing_nothing_wrong.subflow
ESQLs
Java
Test_propagate_loopJava
Maps
Acoount_management_Map_to_audit.map
Acoount_management_Map_to_audit1.map
Test_propagate_loop_Mapping.map

4. If you have already created the WorkloadManagement integration server, you may skip to the next step. Please ensure you are using the correct case, it is important here.

   Create a new integration server (execution group) by right clicking on IB9NODE and selecting .New Integration Server.



   Name the Integration Server WorkloadManagement.

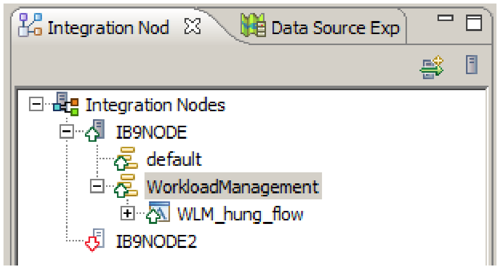


   Click OK.

5. Deploy the application.

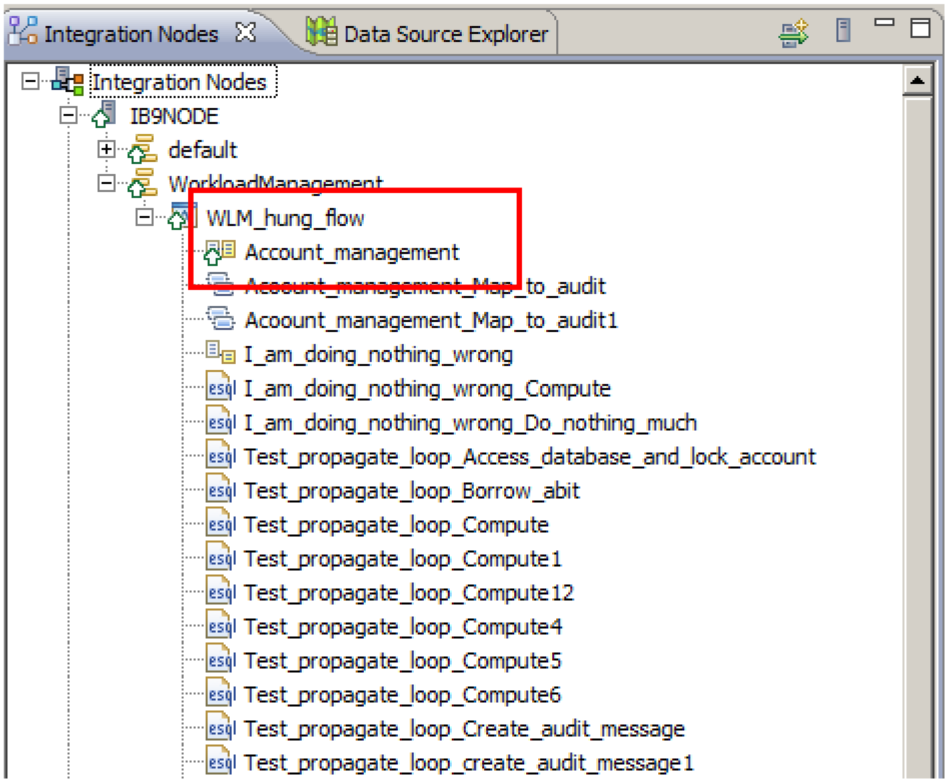
   Click the application **WLM_hung_flow** with the left mouse button and drag and drop it onto the integration server (execution group) **WorkloadManagement**.

   You will now see the application **WLM_hung_flow** running in the integration server **WorkloadManagement**. Notice that it has an up green arrow to indicate it is running. So both the execution group and the application are running.



   Expand the application by clicking the plus sign next to the application.

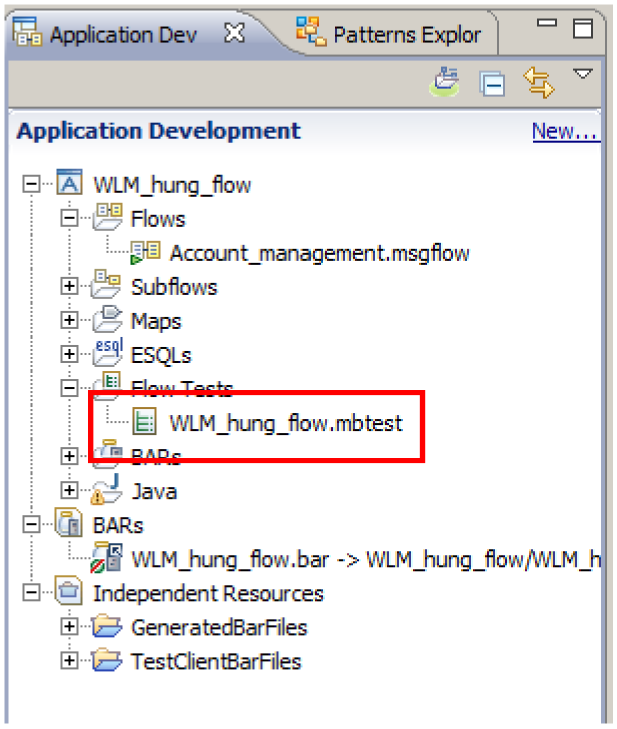
   You now see the message flow Account_management running as well. We are not interested in the other deployed artefacts for the purposes of this lab.


Integration Nodes
Data Source Explorer
Integration Nodes
IB9NODE
default
WorkloadManagement
WLM_hung_flow
Account_management
Account_management_Map_to_audit
Acoount_management_Map_to_audit1
I_am_doing_nothing_wrong
I_am_doing_nothing_wrong_Compute
I_am_doing_nothing_wrong_Do_nothing_much
Test_propagate_loop_Access_database_and_lock_account
Test_propagate_loop_Borrow_abit
Test_propagate_loop_Compute
Test_propagate_loop_Compute1
Test_propagate_loop_Compute12
Test_propagate_loop_Compute4
Test_propagate_loop_Compute5
Test_propagate_loop_Compute6
Test_propagate_loop_Create_audit_message
Test_propagate_loop_create_audit_message1

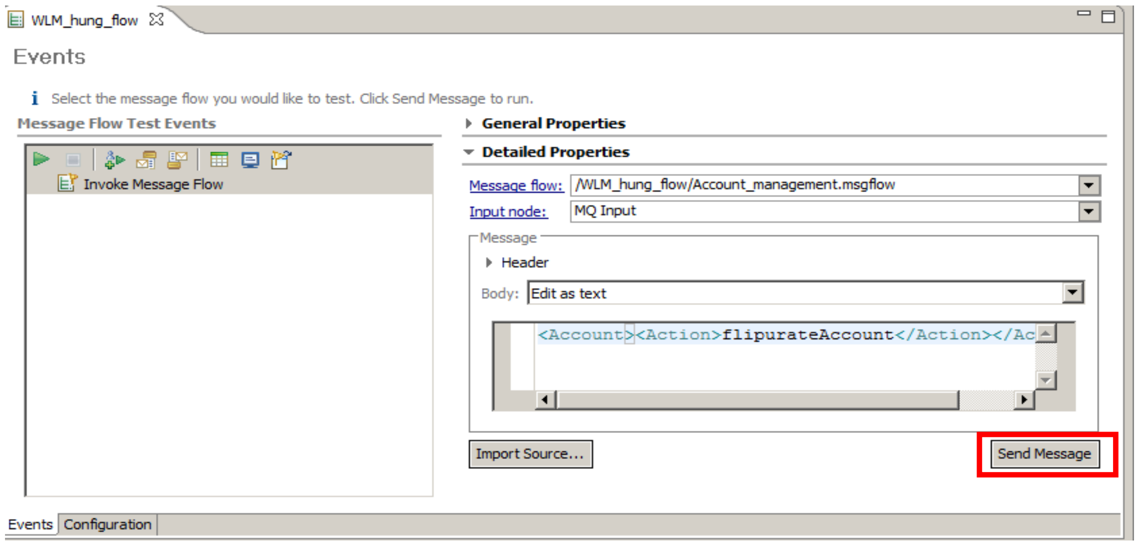
6. Cause the message flows to hang.

   You will now put a message on the MQINPUT node of the flow. This message was purposely constructed to hang the flow in a loop, so it would not stop nor respond to normal mqsistopmsgflow commands from the command console or the toolkit.

   Double click on the WLM_hung_flow.mbtest file under the WLM_hung_flow application, in the folder Flow Tests.



7. Send the message to the Account_management.msgflow by clicking 'Send Message'.



Select the message flow you would like to test. Click Send Message to run.

Message flow: /WLM_hung_flow/Account_management.msgflow

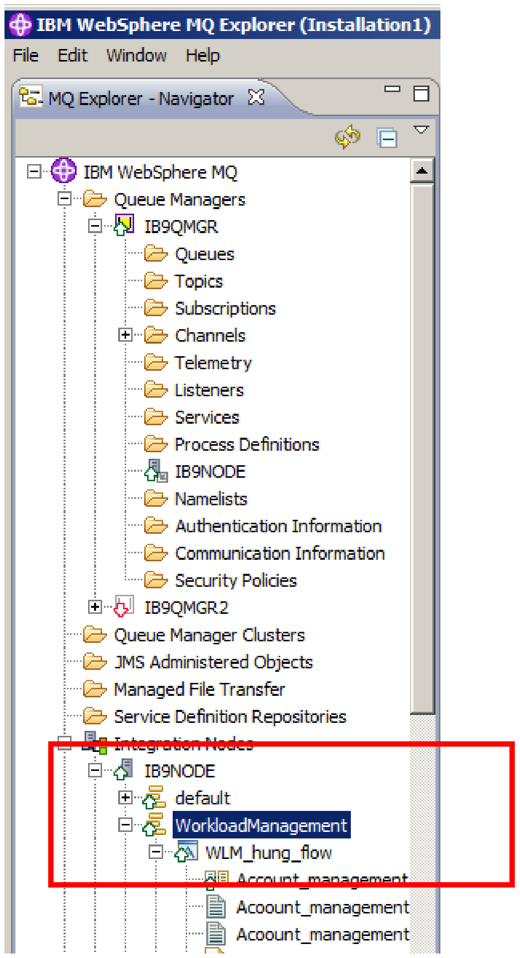
Input node: MQ Input

Message

Body: Edit as text

```
<Account><Action>flipurateAccount</Action></Ac
```

8. Verify that the Account_management flow is still running by viewing IB Explorer or the toolkit.



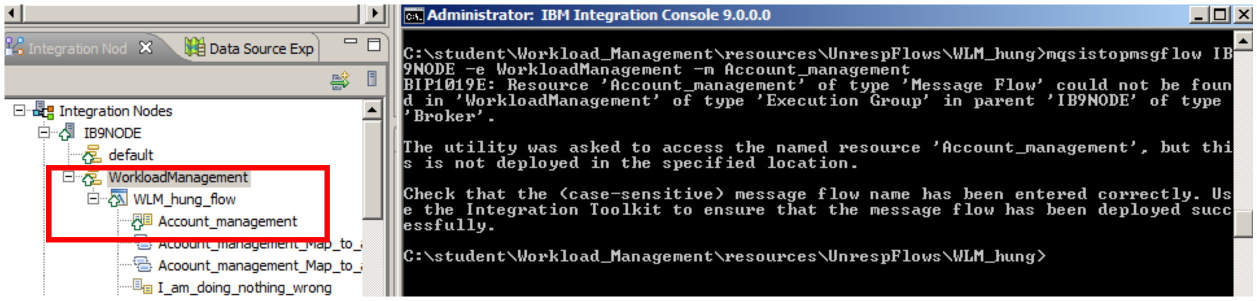
9. Try to stop the message flow.

   Return to the open command console or open it again if you closed it. Make sure it is the Integration Command Console, not a Windows command prompt.

   Try to stop the flow with the following command:

   ```
   mqsistopmsgflow IB9NODE -e WorkloadManagement -m Account_management
   ```

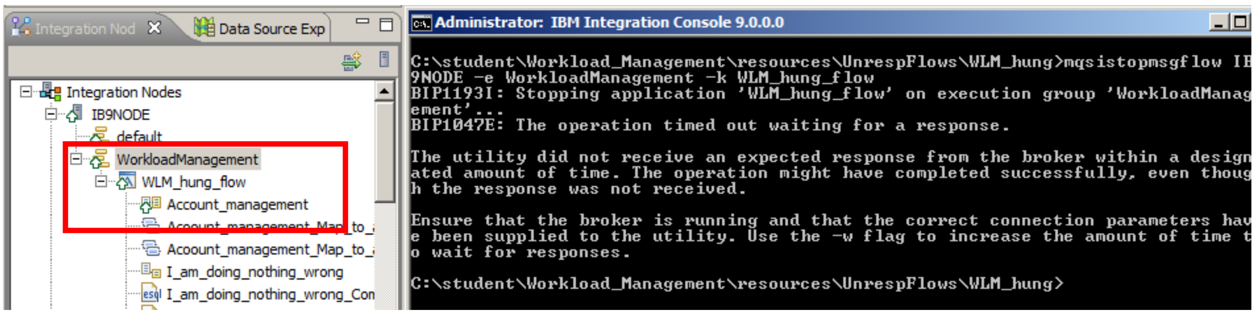
   You will be returned an error message and the flow will continue to run.
   This is because there is a new parameter on the mqsistopmsgflow command; the **–k** option is needed for applications.



```
C:\student\Workload_Management\resources\UnrespFlows\WLM_hung>mqsistopmsgflow IB
9NODE -e WorkloadManagement -m Account_management
BIP1019E: Resource 'Account_management' of type 'Message Flow' could not be foun
d in 'WorkloadManagement' of type 'Execution Group' in parent 'IB9NODE' of type
'Broker'.

The utility was asked to access the named resource 'Account_management', but thi
s is not deployed in the specified location.

Check that the (case-sensitive) message flow name has been entered correctly. Us
e the Integration Toolkit to ensure that the message flow has been deployed succ
essfully.

C:\student\Workload_Management\resources\UnrespFlows\WLM_hung>
```

10. Try stopping the application with the following command:

    ```
    mqsistopmsgflow IB9NODE -e WorkloadManagement -k WLM_hung_flow
    ```

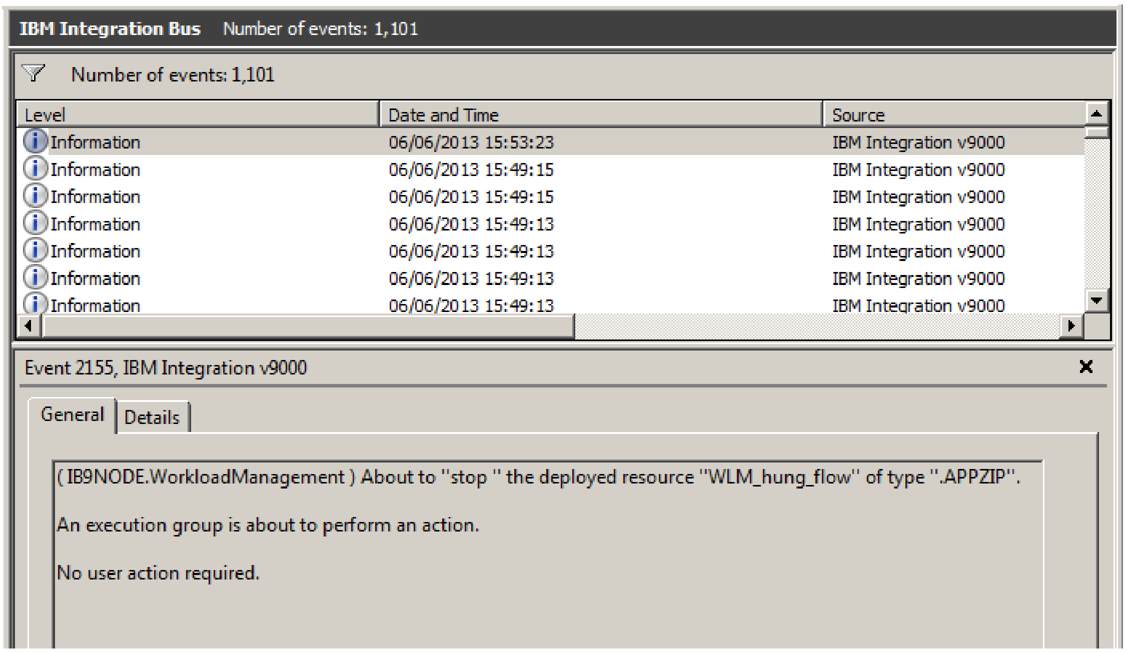


```
C:\student\Workload_Management\resources\UnrespFlows\WLM_hung>mqsistopmsgflow IB
9NODE -e WorkloadManagement -k WLM_hung_flow
BIP1193I: Stopping application 'WLM_hung_flow' on execution group 'WorkloadManag
ement'...
BIP1047E: The operation timed out waiting for a response.

The utility did not receive an expected response from the broker within a design
ated amount of time. The operation might have completed successfully, even thoug
h the response was not received.

Ensure that the broker is running and that the correct connection parameters hav
e been supplied to the utility. Use the -w flag to increase the amount of time t
o wait for responses.

C:\student\Workload_Management\resources\UnrespFlows\WLM_hung>
```

11. Open the Windows Event Log viewer by clicking the icon on the quick start menu.

    Expand Custom Views, and select IBM Integration Bus since we want to view only the Integration Bus messages.

    You will see an information message that the mqsistopmsgflow had been received and the application is about to be stopped.



IBM Integration Bus Number of events: 1,101

Number of events: 1,101

Level	Date and Time	Source
Information	06/06/2013 15:53:23	IBM Integration v9000
Information	06/06/2013 15:49:15	IBM Integration v9000
Information	06/06/2013 15:49:15	IBM Integration v9000
Information	06/06/2013 15:49:13	IBM Integration v9000
Information	06/06/2013 15:49:13	IBM Integration v9000
Information	06/06/2013 15:49:13	IBM Integration v9000
Information	06/06/2013 15:49:13	IBM Integration v9000

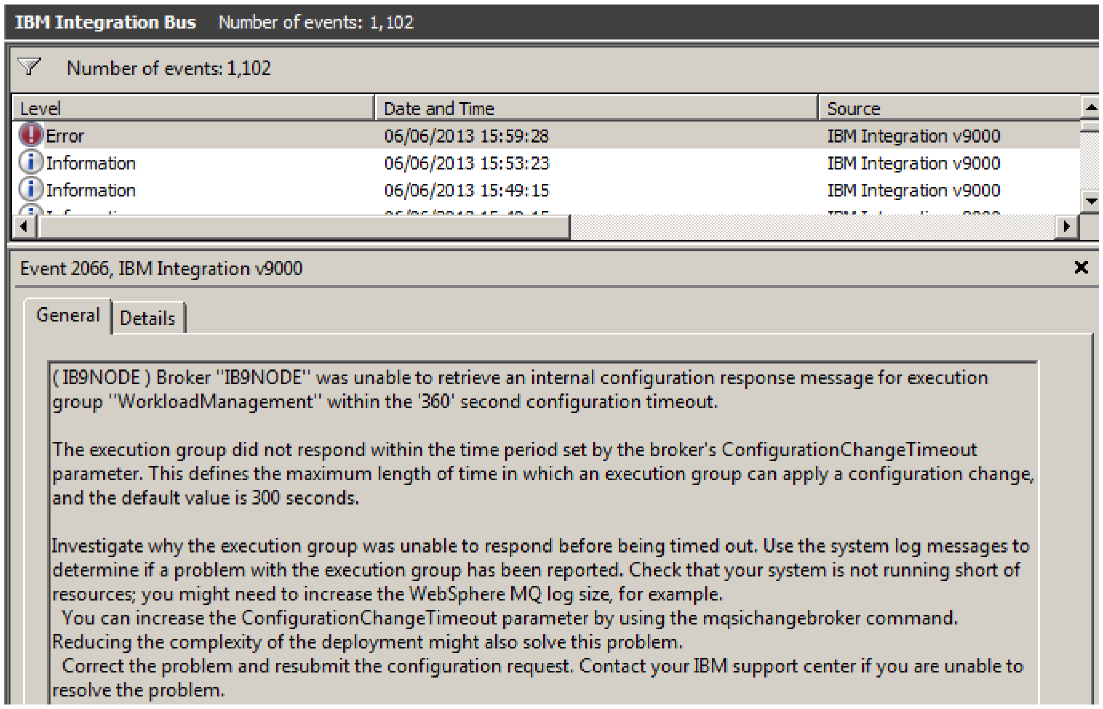
Event 2155, IBM Integration v9000

(IB9NODE.WorkloadManagement) About to ''stop '' the deployed resource ''WLM_hung_flow'' of type ''.APPZIP''.

An execution group is about to perform an action.

No user action required.

12. You should see an error message at a time shortly following the previous message. You may need to refresh the event viewer a number of times.


IBM Integration Bus Number of events: 1,102

Number of events: 1,102

Level	Date and Time	Source
Error	06/06/2013 15:59:28	IBM Integration v9000
Information	06/06/2013 15:53:23	IBM Integration v9000
Information	06/06/2013 15:49:15	IBM Integration v9000

Event 2066, IBM Integration v9000

General Details

(IB9NODE) Broker "IB9NODE" was unable to retrieve an internal configuration response message for execution group "WorkloadManagement" within the '360' second configuration timeout.

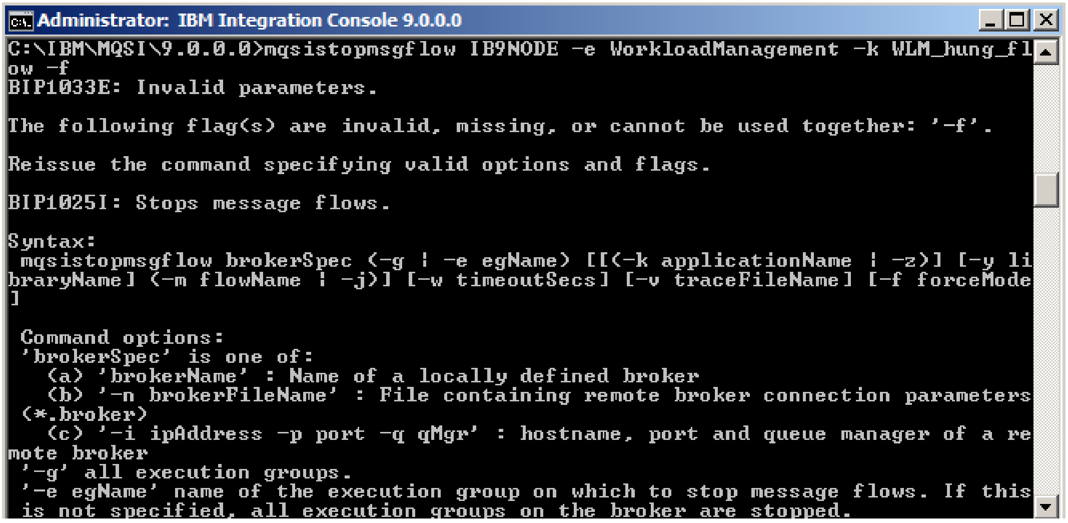
The execution group did not respond within the time period set by the broker's ConfigurationChangeTimeout parameter. This defines the maximum length of time in which an execution group can apply a configuration change, and the default value is 300 seconds.

Investigate why the execution group was unable to respond before being timed out. Use the system log messages to determine if a problem with the execution group has been reported. Check that your system is not running short of resources; you might need to increase the WebSphere MQ log size, for example.
You can increase the ConfigurationChangeTimeout parameter by using the mqsichangebroker command.
Reducing the complexity of the deployment might also solve this problem.
Correct the problem and resubmit the configuration request. Contact your IBM support center if you are unable to resolve the problem.

This appears to be a timeout message because the execution group is not responding because it could not stop the flow. The flow is in fact hung on a looping MQGET node waiting for message to arrive and no timeout set.

13. The –f parameter of the mqsistopmsgflow command now only works with the application. Try to stop the message flow with the new -f parameter. Enter the command:

```
mqsistopmsgflow IB9NODE -e WorkloadManagement -k WLM_hung_flow -f
```

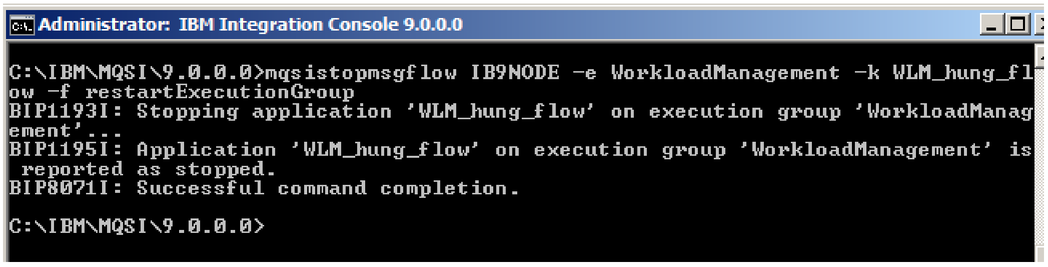

Administrator: IBM Integration Console 9.0.0.0

```
C:\IBM\MQSI\9.0.0.0>mqsistopmsgflow IB9NODE -e WorkloadManagement -k WLM_hung_fl
ow -f
BIP1033E: Invalid parameters.

The following flag(s) are invalid, missing, or cannot be used together: '-f'.

Reissue the command specifying valid options and flags.

BIP1025I: Stops message flows.

Syntax:
 mqsistopmsgflow brokerSpec (-g | -e egName) [[(-k applicationName | -z)] [-y li
braryName] (-m flowName | -j)] [-w timeoutSecs] [-v traceFileName] [-f forceMode
]

 Command options:
 'brokerSpec' is one of:
   (a) 'brokerName' : Name of a locally defined broker
   (b) '-n brokerFileName' : File containing remote broker connection parameters
 (*.broker)
   (c) '-i ipAddress -p port -q qMgr' : hostname, port and queue manager of a re
mote broker
 '-g' all execution groups.
 '-e egName' name of the execution group on which to stop message flows. If this
 is not specified, all execution groups on the broker are stopped.
```

This error message is intentional, because the command will only work with the new force parameter **–f restartExecutionGroup**. The –f parameter of the mqsistopmsgflow command now only works with the application name **and** the parameter restartExecutionGroup.

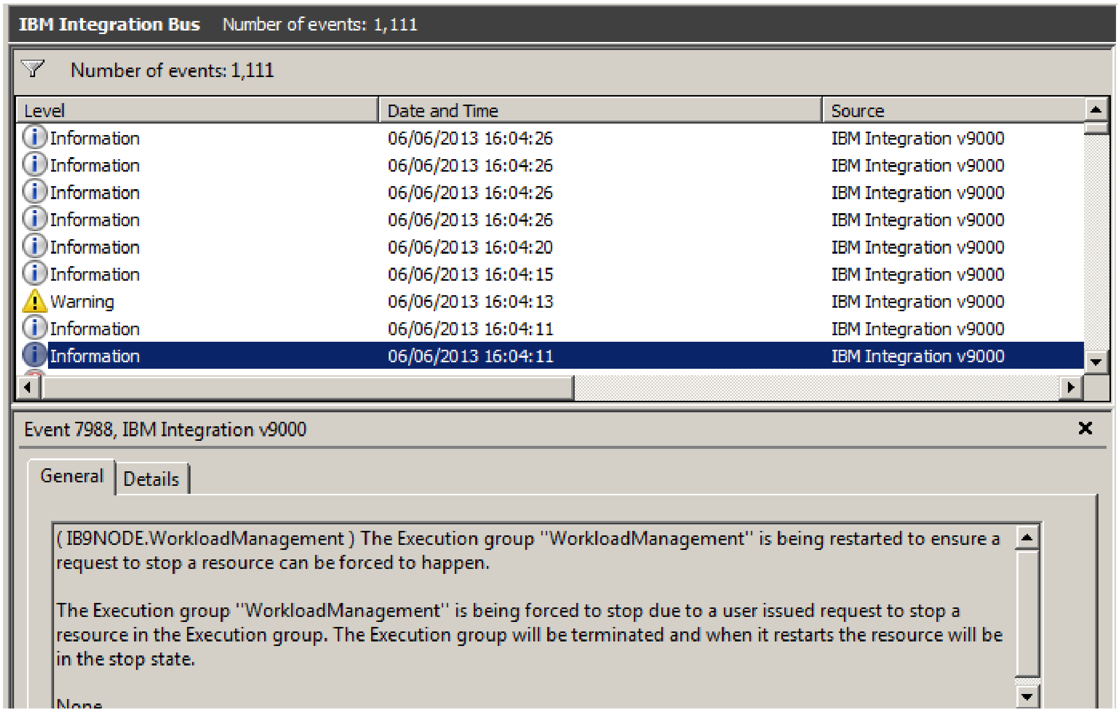
14. Enter the mqsistopmsgflow command one more time. This time add the parameter restartExecutionGroup.

```
mqsistopmsgflow IB9NODE
  -e WorkloadManagement
  -k WLM_hung_flow
  -f restartExecutionGroup
```



```
C:\IBM\MQSI\9.0.0.0>mqsistopmsgflow IB9NODE -e WorkloadManagement -k WLM_hung_fl
ow -f restartExecutionGroup
BIP1193I: Stopping application 'WLM_hung_flow' on execution group 'WorkloadManag
ement'...
BIP1195I: Application 'WLM_hung_flow' on execution group 'WorkloadManagement' is
 reported as stopped.
BIP8071I: Successful command completion.

C:\IBM\MQSI\9.0.0.0>
```

15. Refresh the Event Viewer by clicking the Refresh tab to view additional messages.



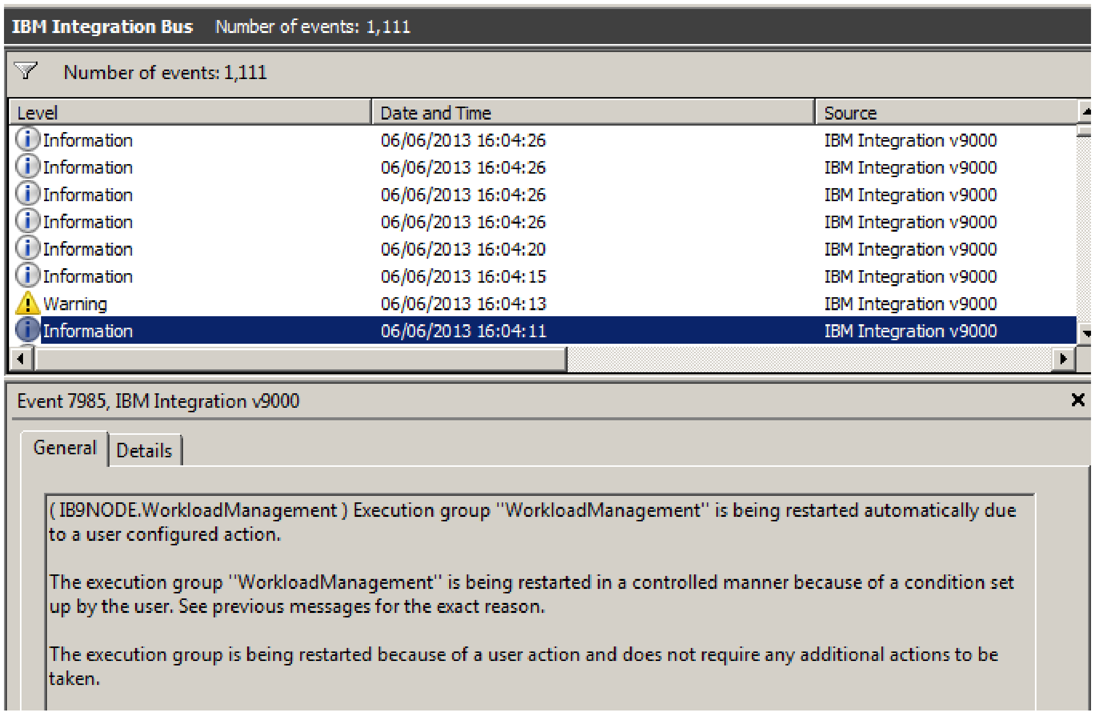
Level	Date and Time	Source
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:20	IBM Integration v9000
Information	06/06/2013 16:04:15	IBM Integration v9000
Warning	06/06/2013 16:04:13	IBM Integration v9000
Information	06/06/2013 16:04:11	IBM Integration v9000
Information	06/06/2013 16:04:11	IBM Integration v9000

(IB9NODE.WorkloadManagement) The Execution group "WorkloadManagement" is being restarted to ensure a request to stop a resource can be forced to happen.

The Execution group "WorkloadManagement" is being forced to stop due to a user issued request to stop a resource in the Execution group. The Execution group will be terminated and when it restarts the resource will be in the stop state.

None

16. You should several information messages at the time you issued the mqsistopmsgflow force command. Start with the first message above the error message and reach each of the succeeding messages to follow what happened due to the force command.



IBM Integration Bus Number of events: 1,111

Number of events: 1,111

Level	Date and Time	Source
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:20	IBM Integration v9000
Information	06/06/2013 16:04:15	IBM Integration v9000
Warning	06/06/2013 16:04:13	IBM Integration v9000
Information	06/06/2013 16:04:11	IBM Integration v9000

Event 7985, IBM Integration v9000

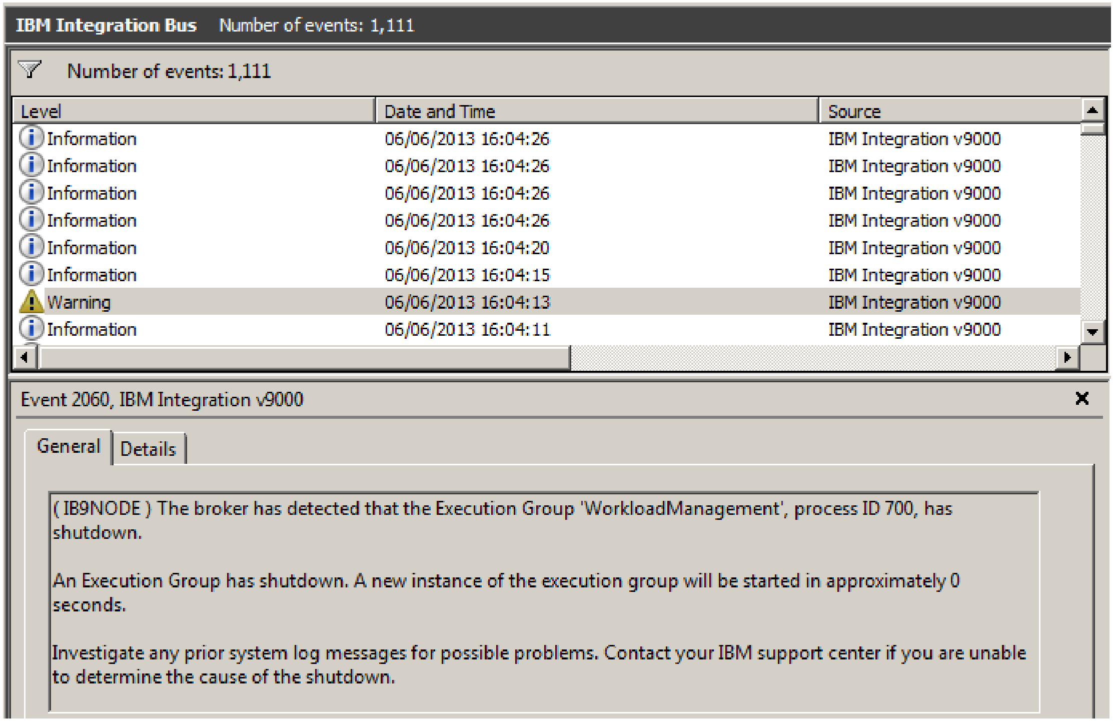
General | Details

(IB9NODE.WorkloadManagement) Execution group ''WorkloadManagement'' is being restarted automatically due to a user configured action.

The execution group ''WorkloadManagement'' is being restarted in a controlled manner because of a condition set up by the user. See previous messages for the exact reason.

The execution group is being restarted because of a user action and does not require any additional actions to be taken.

Informational message - the execution group is being restarted due to a user configured action. Next is a warning message that the execution group has shutdown.



IBM Integration Bus Number of events: 1,111

Number of events: 1,111

Level	Date and Time	Source
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:20	IBM Integration v9000
Information	06/06/2013 16:04:15	IBM Integration v9000
Warning	06/06/2013 16:04:13	IBM Integration v9000
Information	06/06/2013 16:04:11	IBM Integration v9000

Event 2060, IBM Integration v9000

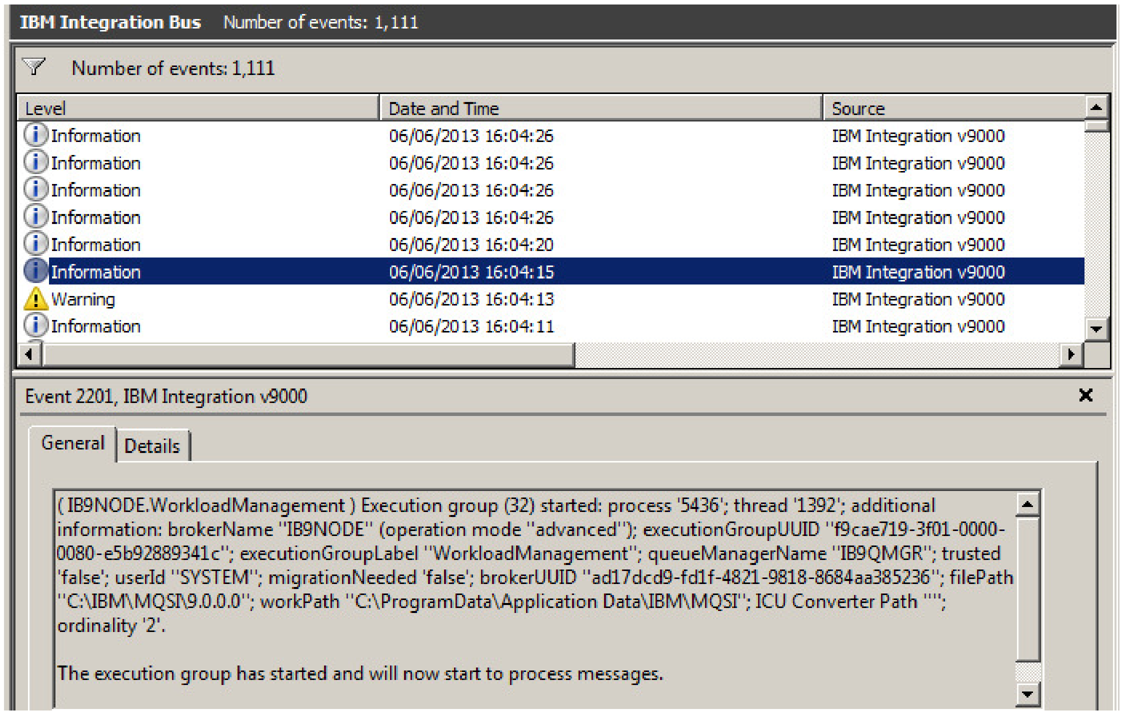
General | Details

(IB9NODE) The broker has detected that the Execution Group 'WorkloadManagement', process ID 700, has shutdown.

An Execution Group has shutdown. A new instance of the execution group will be started in approximately 0 seconds.

Investigate any prior system log messages for possible problems. Contact your IBM support center if you are unable to determine the cause of the shutdown.

Succeeding the warning message, there are five informational messages explaining what is being done. The first informational message shows that the execution group has restarted. This ensures that other flows keep running and are not impacted by the stop command for the hung flow.



IBM Integration Bus Number of events: 1,111

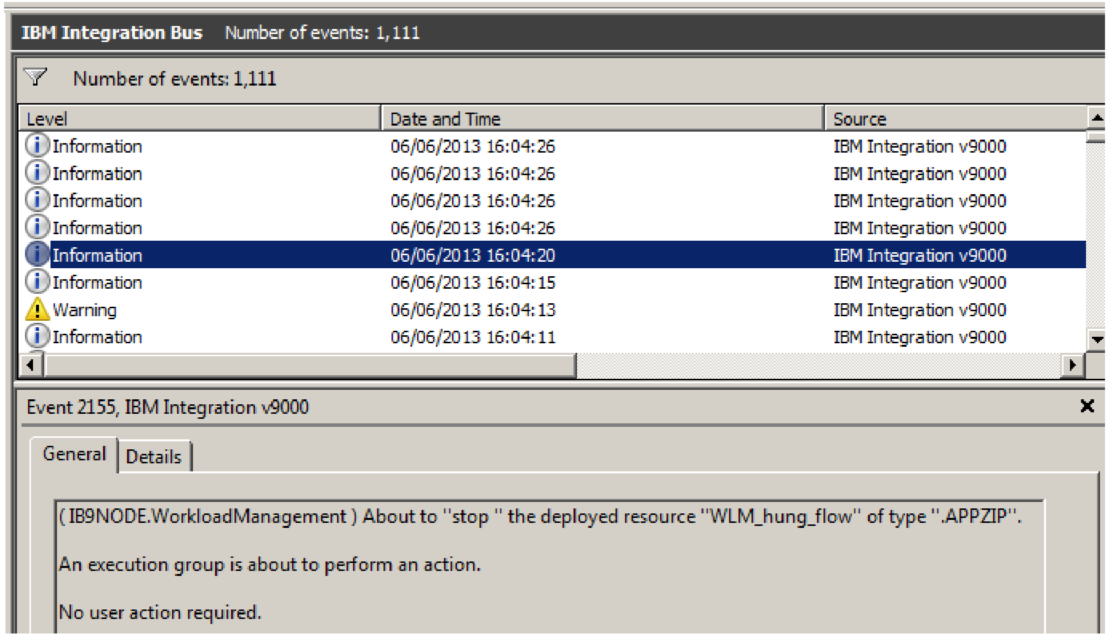
Number of events: 1,111

Level	Date and Time	Source
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:20	IBM Integration v9000
Information	06/06/2013 16:04:15	IBM Integration v9000
Warning	06/06/2013 16:04:13	IBM Integration v9000
Information	06/06/2013 16:04:11	IBM Integration v9000

Event 2201, IBM Integration v9000

```
( IB9NODE.WorkloadManagement ) Execution group (32) started: process '5436'; thread '1392'; additional information: brokerName "IB9NODE" (operation mode "advanced"); executionGroupUUID "f9cae719-3f01-0000-0080-e5b92889341c"; executionGroupLabel "WorkloadManagement"; queueManagerName "IB9QMGR"; trusted 'false'; userId "SYSTEM"; migrationNeeded 'false'; brokerUUID "ad17dcd9-fd1f-4821-9818-8684aa385236"; filePath "C:\IBM\MQSI\9.0.0.0"; workPath "C:\ProgramData\Application Data\IBM\MQSI"; ICU Converter Path "";
ordinality '2'.

The execution group has started and will now start to process messages.
```

Next informational message shows stopping the flow which is hung.



IBM Integration Bus Number of events: 1,111

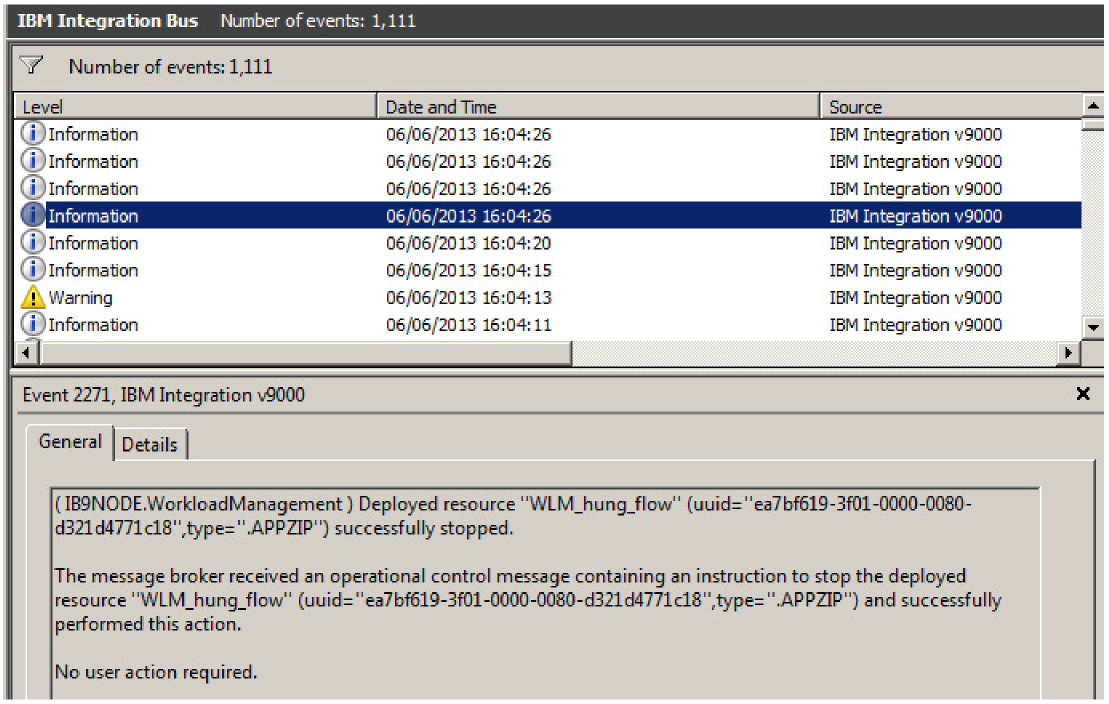
Number of events: 1,111

Level	Date and Time	Source
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:20	IBM Integration v9000
Information	06/06/2013 16:04:15	IBM Integration v9000
Warning	06/06/2013 16:04:13	IBM Integration v9000
Information	06/06/2013 16:04:11	IBM Integration v9000

Event 2155, IBM Integration v9000

```
( IB9NODE.WorkloadManagement ) About to ''stop '' the deployed resource ''WLM_hung_flow'' of type ''.APPZIP''.

An execution group is about to perform an action.

No user action required.
```

The next message shows that the flow was indeed stopped.

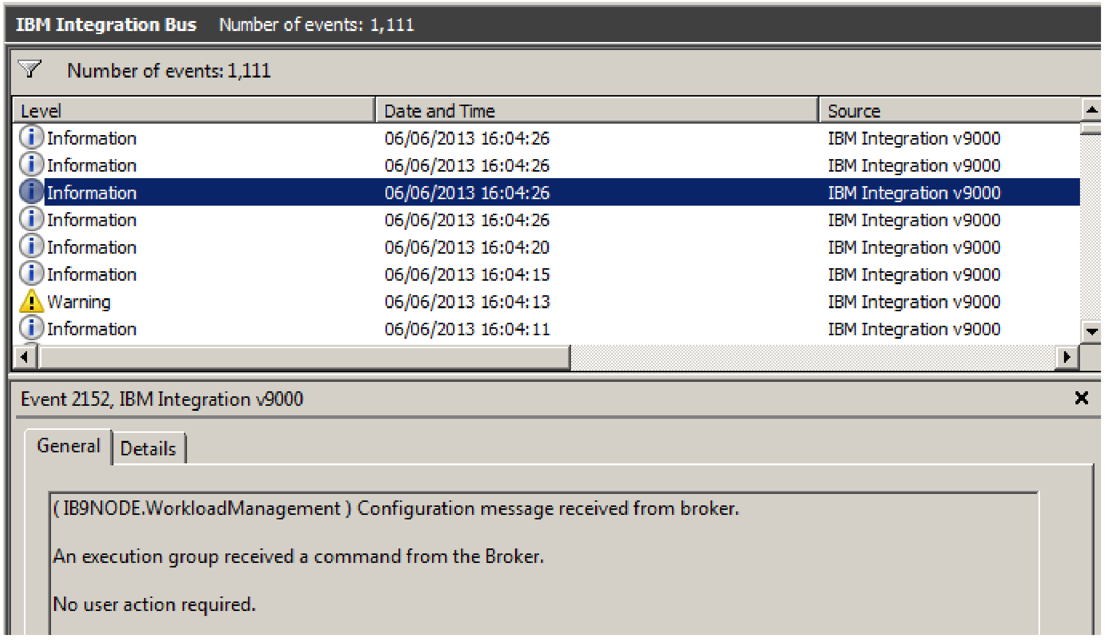


Level	Date and Time	Source
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:20	IBM Integration v9000
Information	06/06/2013 16:04:15	IBM Integration v9000
Warning	06/06/2013 16:04:13	IBM Integration v9000
Information	06/06/2013 16:04:11	IBM Integration v9000

Event 2271, IBM Integration v9000

```
( IB9NODE.WorkloadManagement ) Deployed resource "WLM_hung_flow" (uuid="ea7bf619-3f01-0000-0080-d321d4771c18",type=".APPZIP") successfully stopped.

The message broker received an operational control message containing an instruction to stop the deployed resource "WLM_hung_flow" (uuid="ea7bf619-3f01-0000-0080-d321d4771c18",type=".APPZIP") and successfully performed this action.

No user action required.
```

Further informational messages shows the broker sending messages to the execution group.

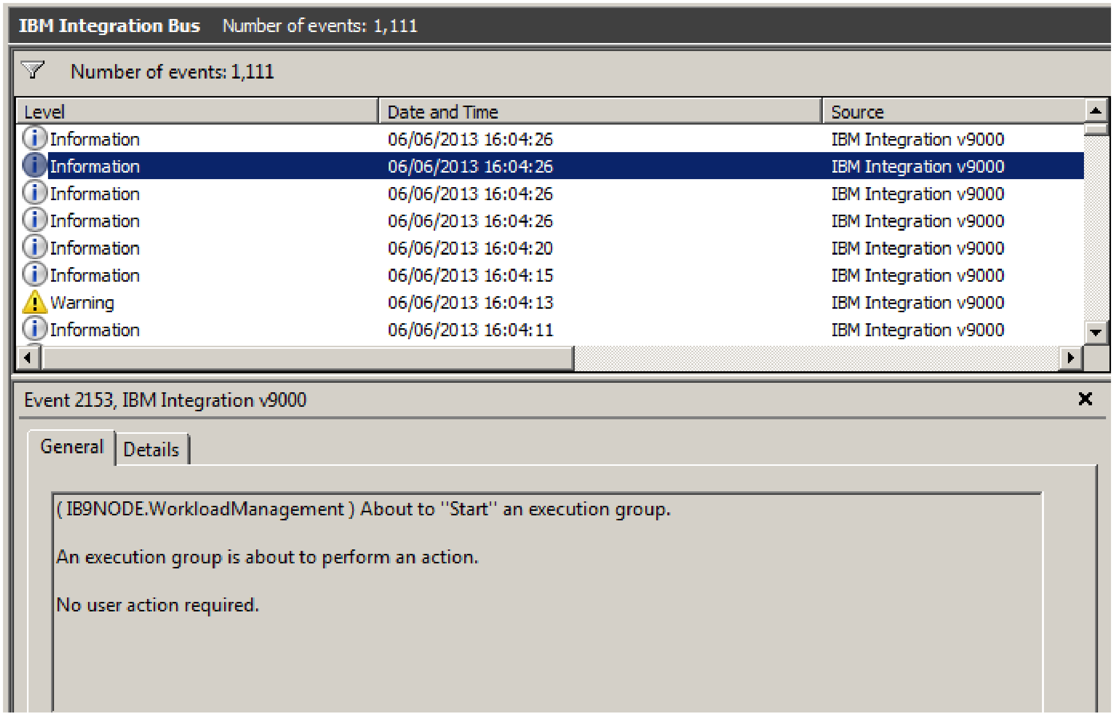


Level	Date and Time	Source
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:20	IBM Integration v9000
Information	06/06/2013 16:04:15	IBM Integration v9000
Warning	06/06/2013 16:04:13	IBM Integration v9000
Information	06/06/2013 16:04:11	IBM Integration v9000

Event 2152, IBM Integration v9000

```
( IB9NODE.WorkloadManagement ) Configuration message received from broker.

An execution group received a command from the Broker.

No user action required.
```

And finally the execution group has started.


IBM Integration Bus Number of events: 1,111

Number of events: 1,111

Level	Date and Time	Source
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:26	IBM Integration v9000
Information	06/06/2013 16:04:20	IBM Integration v9000
Information	06/06/2013 16:04:15	IBM Integration v9000
Warning	06/06/2013 16:04:13	IBM Integration v9000
Information	06/06/2013 16:04:11	IBM Integration v9000

Event 2153, IBM Integration v9000

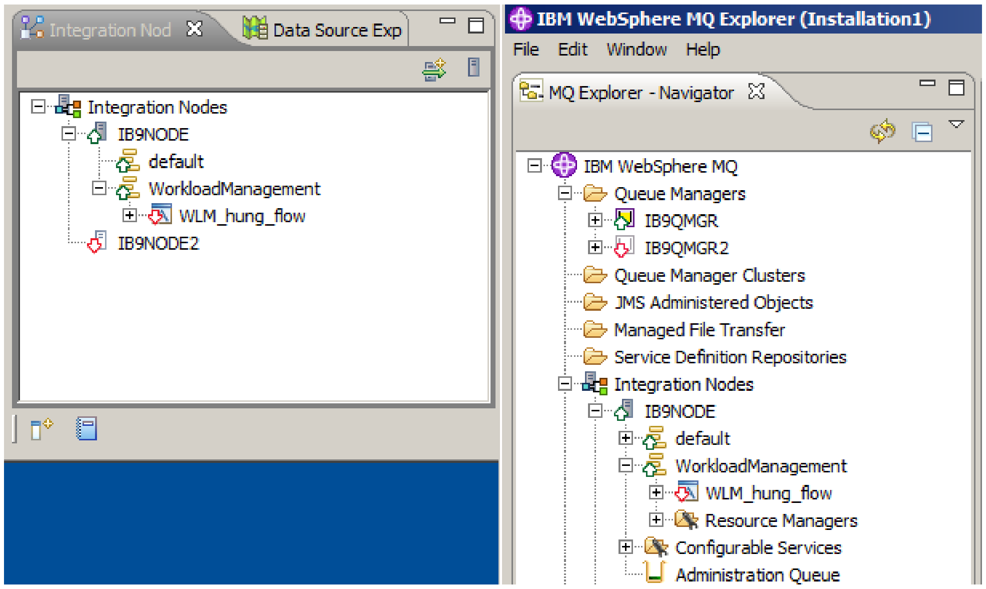
General Details

(IB9NODE.WorkloadManagement) About to "Start" an execution group.

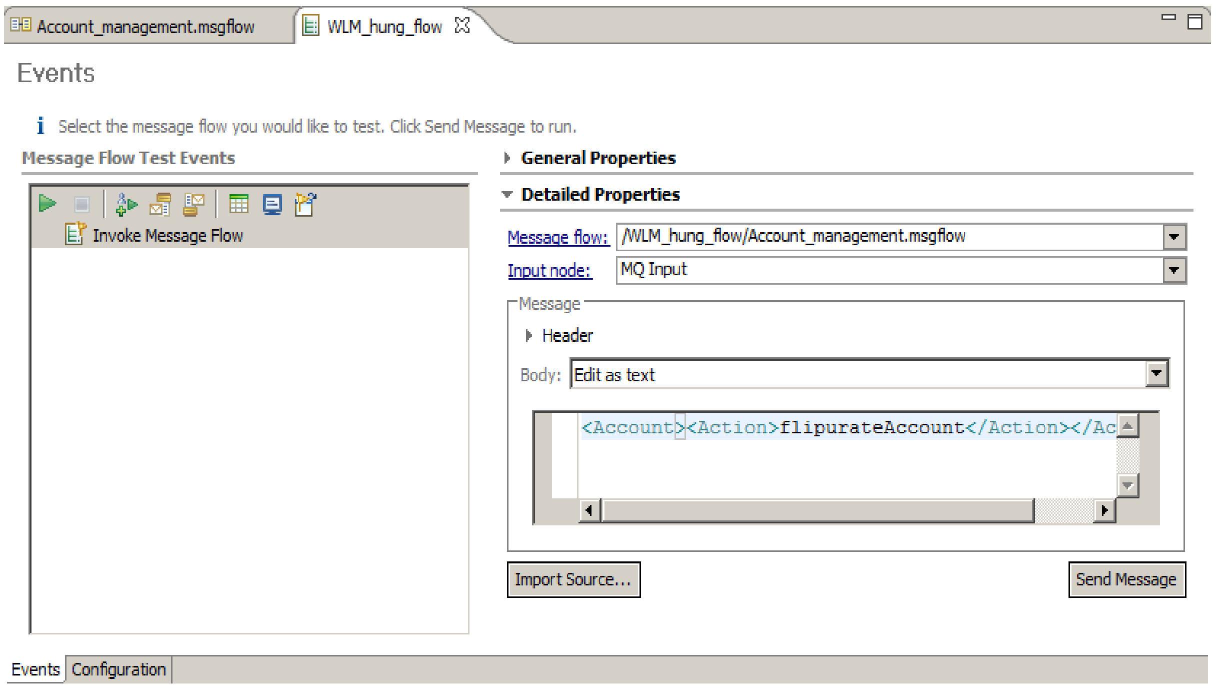
An execution group is about to perform an action.

No user action required.

Returning to the IIB Explorer in MQ Explorer or in the IIB V9 Toolkit, you will see the execution group 'WorkloadManagement' is started, but the application WLM_hung_flow which contains the culprit flow Account_management is stopped.



17. Close the WLM_hung_flow test window but **do not** save the test.



Message flow: /WLM_hung_flow/Account_management.msgflow
Input node: MQ Input
Body: Edit as text

```
<Account><Action>flipurateAccount</Action></Ac
```

4. Automatically force a message flow to stop

Using a policy in the properties of a message flow, you can now monitor message flow processing time and automatically take a specified action if the time-out value is exceeded.

Two message flow properties are provided to specify the maximum amount of time that any message flow can be allowed to process a message and an action to be invoked if the timeout is exceeded:

- processingTimeoutSec – maximum time a message flow can process a message before taking a specified action. The time is measured in seconds and is taken from the point a message is received on an input node.
- processingTimeoutAction – the action to take. Currently, this action is restricted to none or restartExecutionGroup.

Additionally, an event message is published on a WebSphere MQ topic once the processingTimeoutSec is exceeded and again when the message flow processing has finished.

There are two ways that both of these properties can be set for a message flow:

- Directly within a BAR file.
- As one of the attributes within a workload management policy that is defined within a configurable service.

4.1 Setting timeout properties in the barfile

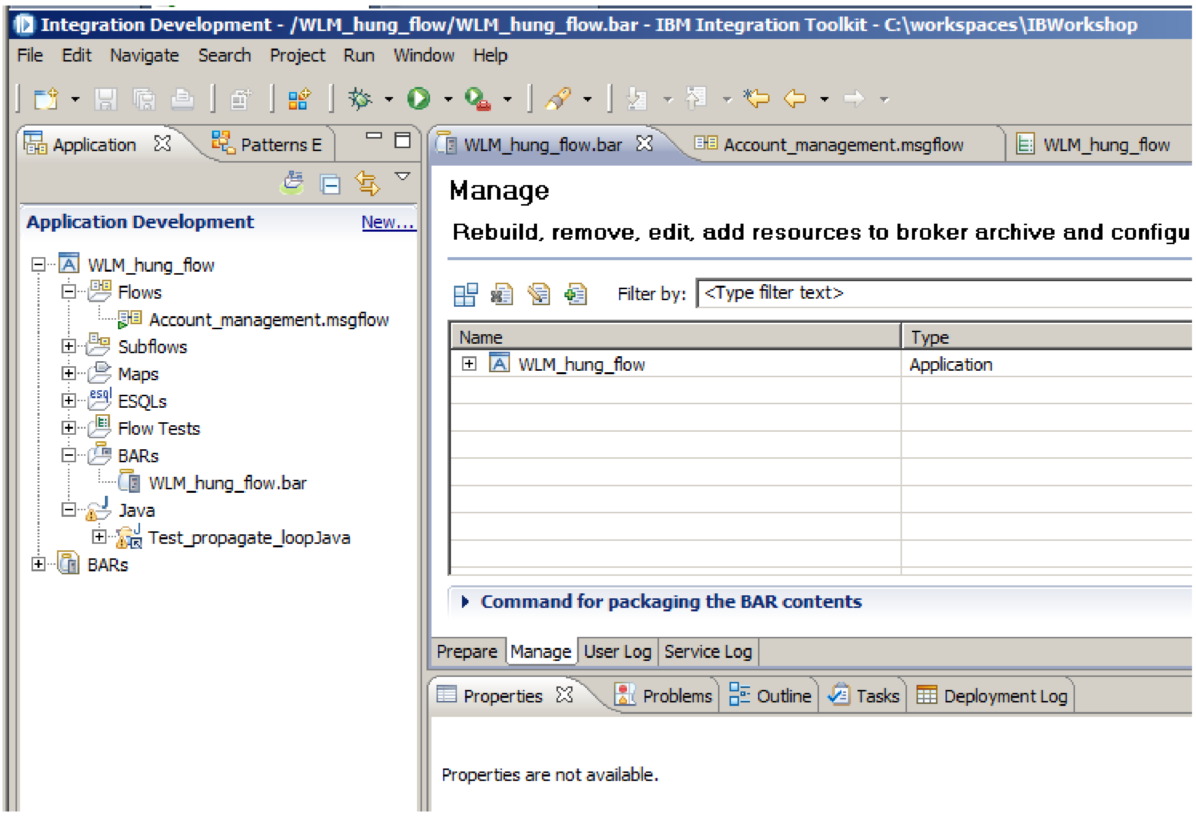
The two properties can be set in the following ways:

- With the Integration Toolkit editor.
- With barfile editor in the IBM Integration Explorer.
- With the mqsiapplybaroverride command line.

In this lab, we will demonstrate the barfile editor in the Integration Toolkit.

1. In the Integration Toolkit, open the barfile WLM_hung_flow.bar (in the WLM_hung_flow application).

   Click the Manage tab.


Integration Development - /WLM_hung_flow/WLM_hung_flow.bar - IBM Integration Toolkit - C:\workspaces\IBWorkshop

Manage

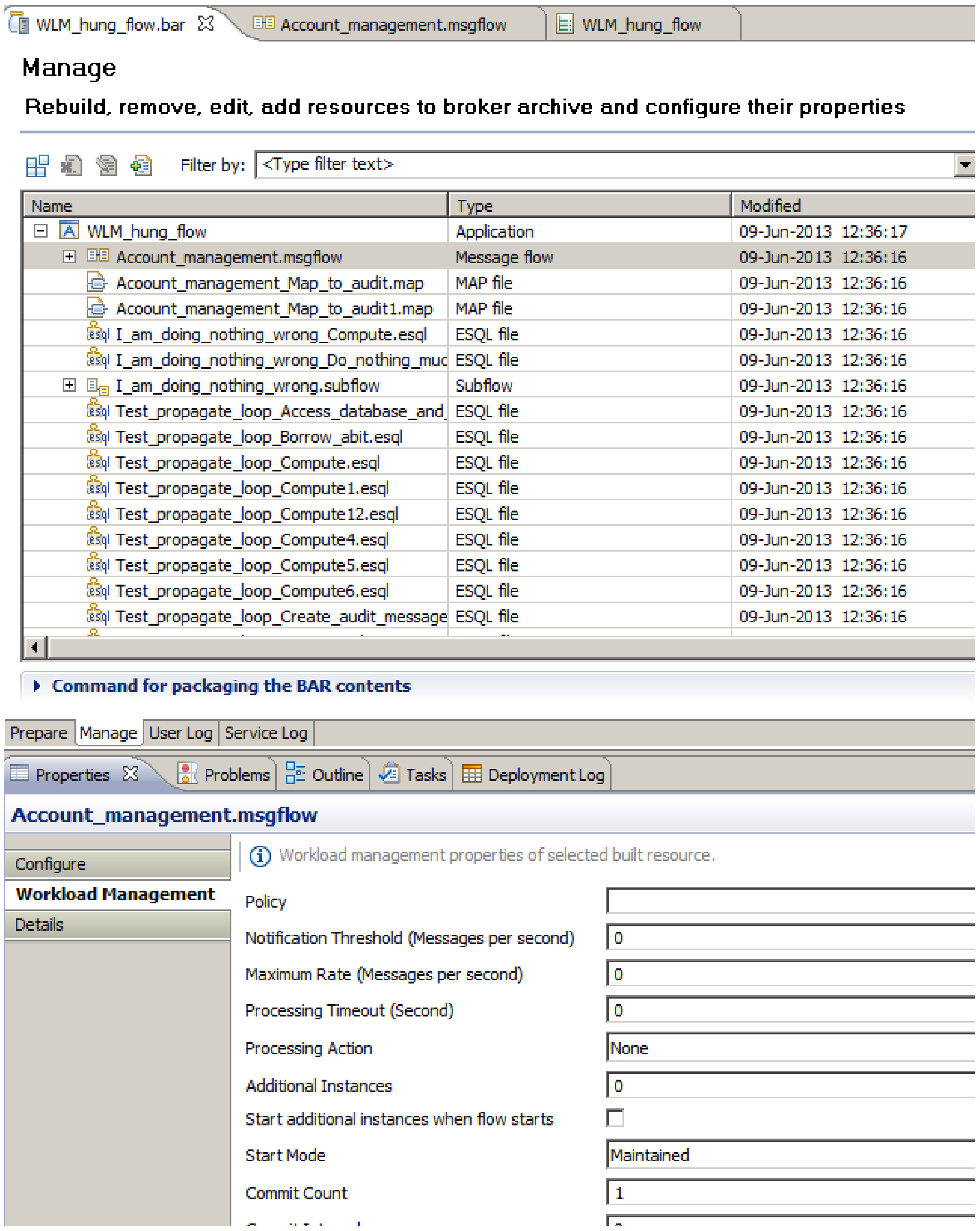
Rebuild, remove, edit, add resources to broker archive and configu

Filter by: <Type filter text>

Name	Type
WLM_hung_flow	Application

Properties are not available.

2. Expand the application in the edit pane by clicking on the + sign.
Click the Account_management.msgflow to select it and select Workload Management in the Properties view.



WLM_hung_flow.bar | Account_management.msgflow | WLM_hung_flow

Manage

Rebuild, remove, edit, add resources to broker archive and configure their properties

Filter by: <Type filter text>

Name	Type	Modified
WLM_hung_flow	Application	09-Jun-2013 12:36:17
Account_management.msgflow	Message flow	09-Jun-2013 12:36:16
Acoount_management_Map_to_audit.map	MAP file	09-Jun-2013 12:36:16
Acoount_management_Map_to_audit1.map	MAP file	09-Jun-2013 12:36:16
I_am_doing_nothing_wrong_Compute.esql	ESQL file	09-Jun-2013 12:36:16
I_am_doing_nothing_wrong_Do_nothing_muc	ESQL file	09-Jun-2013 12:36:16
I_am_doing_nothing_wrong.subflow	Subflow	09-Jun-2013 12:36:16
Test_propagate_loop_Access_database_and_	ESQL file	09-Jun-2013 12:36:16
Test_propagate_loop_Borrow_abit.esql	ESQL file	09-Jun-2013 12:36:16
Test_propagate_loop_Compute.esql	ESQL file	09-Jun-2013 12:36:16
Test_propagate_loop_Compute1.esql	ESQL file	09-Jun-2013 12:36:16
Test_propagate_loop_Compute12.esql	ESQL file	09-Jun-2013 12:36:16
Test_propagate_loop_Compute4.esql	ESQL file	09-Jun-2013 12:36:16
Test_propagate_loop_Compute5.esql	ESQL file	09-Jun-2013 12:36:16
Test_propagate_loop_Compute6.esql	ESQL file	09-Jun-2013 12:36:16
Test_propagate_loop_Create_audit_message	ESQL file	09-Jun-2013 12:36:16

Command for packaging the BAR contents

Prepare | Manage | User Log | Service Log

Properties | Problems | Outline | Tasks | Deployment Log

Account_management.msgflow

Configure | **Workload Management** | Details

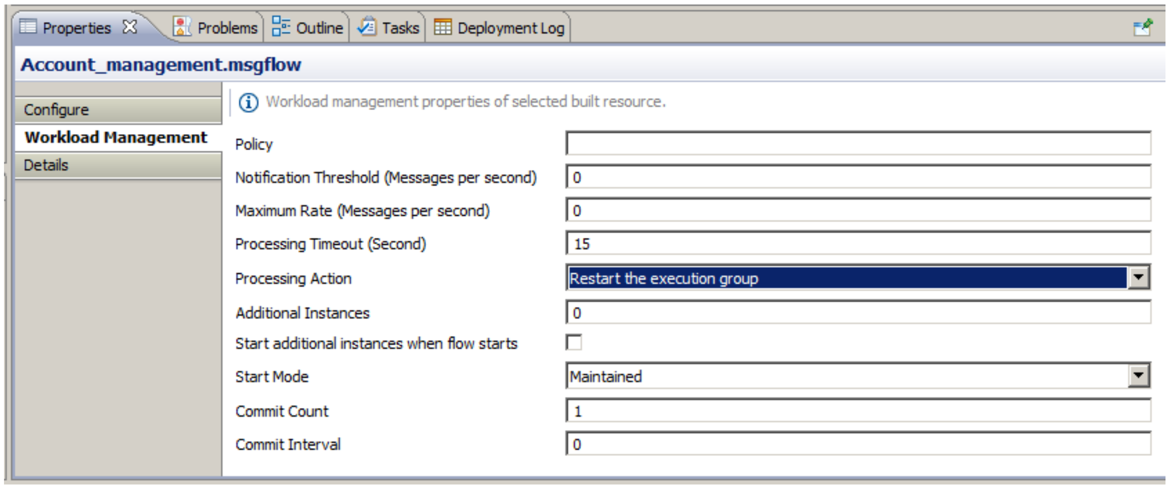
Workload management properties of selected built resource.

Policy	
Notification Threshold (Messages per second)	0
Maximum Rate (Messages per second)	0
Processing Timeout (Second)	0
Processing Action	None
Additional Instances	0
Start additional instances when flow starts	☐
Start Mode	Maintained
Commit Count	1

3. It is in the Workload Management properties where you will find the properties for automatically stopping a flow. The properties we need are Processing Timeout and Processing Action.

   In the Processing Timeout field, type 15 which is the number of seconds a flow will be allowed to process before it is automatically stopped.

   In the Processing Action field, click the down arrow to get the pull-down menu, then select “Restart the execution group”.



Account_management.msgflow

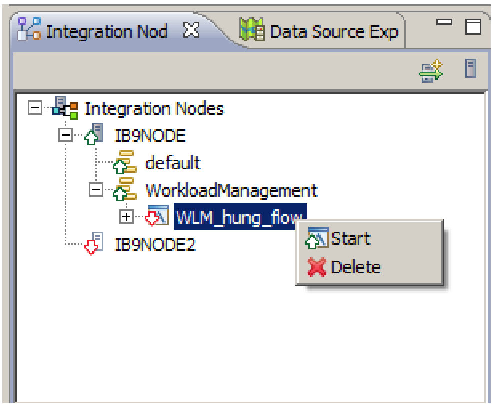
Workload management properties of selected built resource.

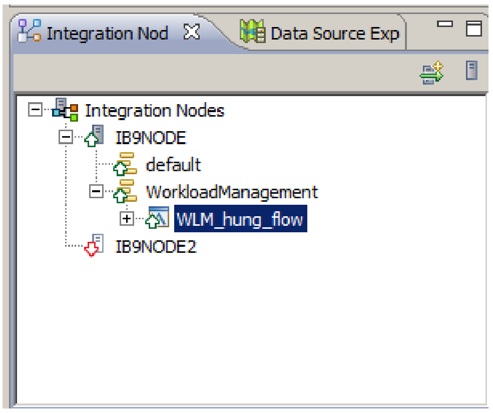
Policy	
Notification Threshold (Messages per second)	0
Maximum Rate (Messages per second)	0
Processing Timeout (Second)	15
Processing Action	Restart the execution group
Additional Instances	0
Start additional instances when flow starts	☐
Start Mode	Maintained
Commit Count	1
Commit Interval	0

   Press CTRL + S to save the properties in the BAR file.

4. Re-deploy the barfile in the usual way, deploying to WorkloadManagement server.

5. Restart the application WLM_hung_flow by right clicking on it and select Start.



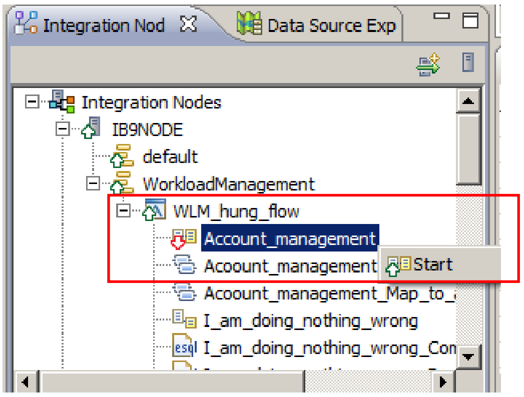


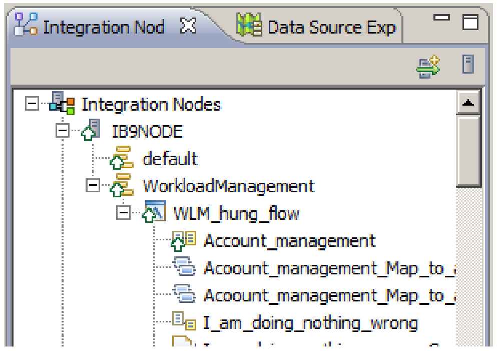
   The application is now started.

6. Even though the application was started, please verify that the message flow Account_management is also started. Expand the application to show the flow. Notice that is still stopped because of the explicit command to stop it in the previous section.

   We are assuming that some corrective action was taken to correct the errors. So now the flow must be explicitly started.

   Right click on Account_management and select Start.



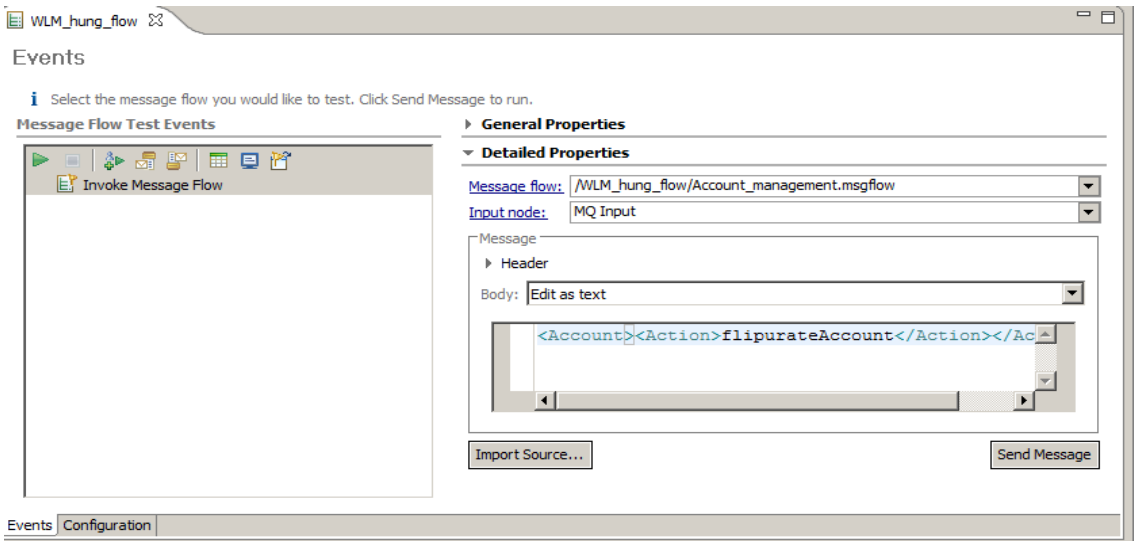


The flow is now started.

7. Now that the flow is redeployed with new properties, we want to hang it again and check to make sure it is automatically removed from the system.

   Hang the flow.
   Now you will put a message on the MQINPUT node of the flow. This message was purposely constructed to hang the flow in a loop, so it would stop nor respond to normal mqsistopmsgflow commands from the command console or the toolkit.

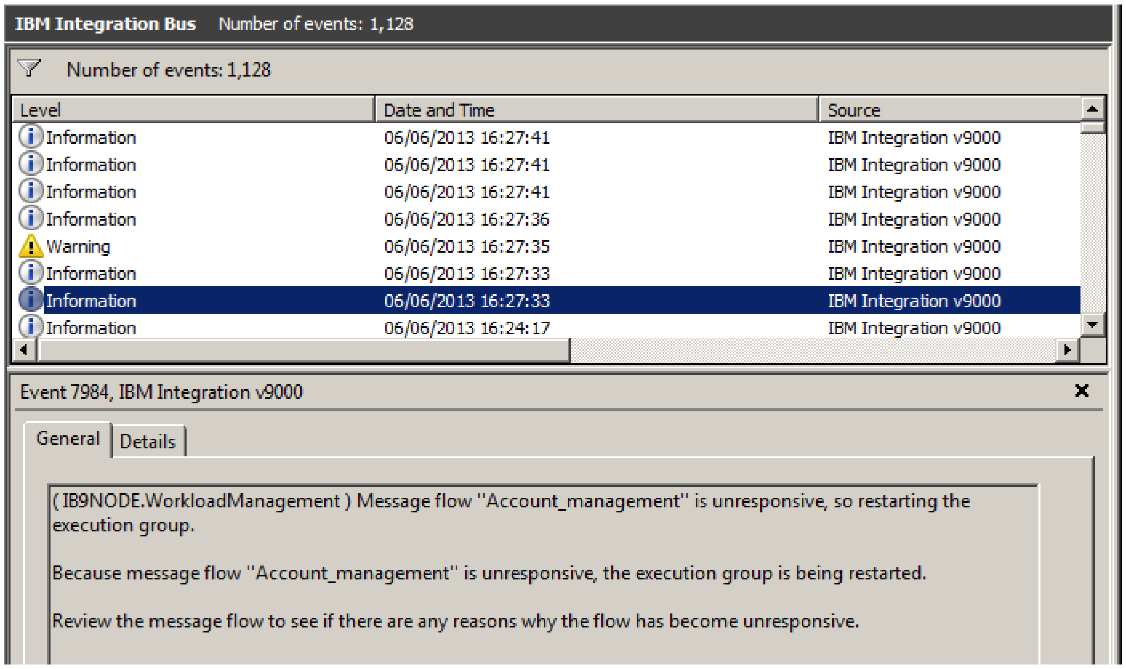
   Double click on the WLM_hung_flow.mbtest file under the WLM_hung_flow application, in the folder Flow Tests. Click send message as before.



Message flow: /WLM_hung_flow/Account_management.msgflow
Input node: MQ Input
Body: Edit as text

```
<Account><Action>flipurateAccount</Action></Ac
```

8. Return to the Event Viewer and click on Refresh. You will see seven new messages at approximately the time you wrote the message to the queue IN1 in RFHUtil. You may only see four initially. You may need to wait and click Refresh a number times. There will be one warning message and six informational messages. Click on the first and observe the message.



Level	Date and Time	Source
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:36	IBM Integration v9000
Warning	06/06/2013 16:27:35	IBM Integration v9000
Information	06/06/2013 16:27:33	IBM Integration v9000
Information	06/06/2013 16:27:33	IBM Integration v9000
Information	06/06/2013 16:24:17	IBM Integration v9000

Event 7984, IBM Integration v9000

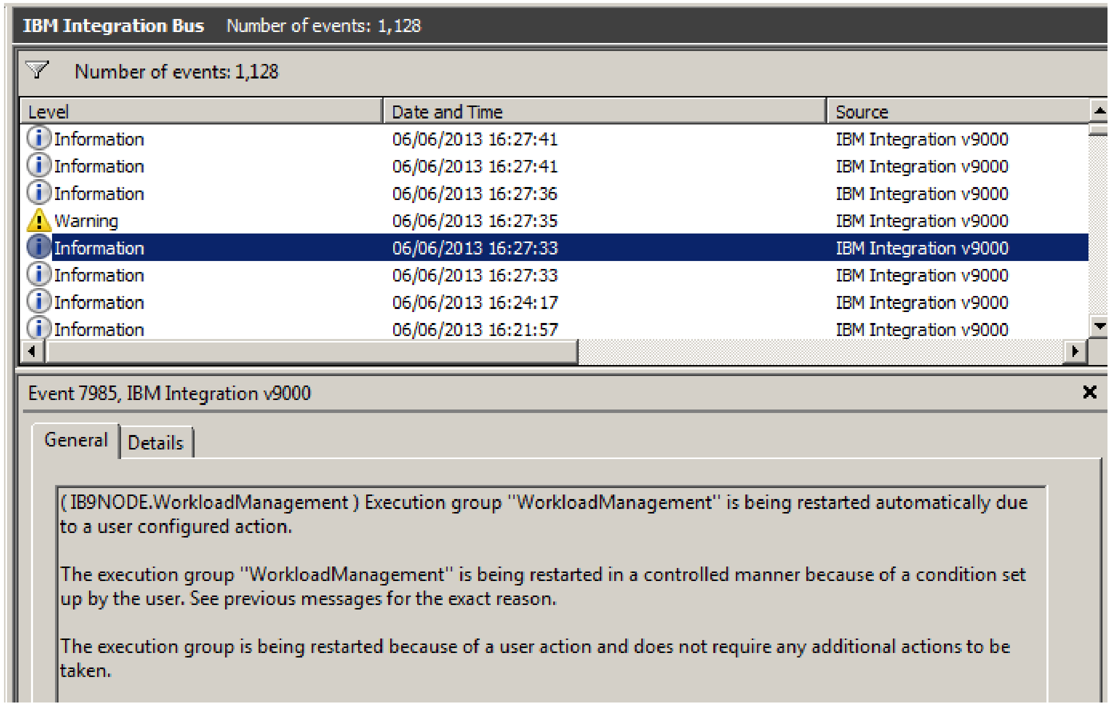
(IB9NODE.WorkloadManagement) Message flow "Account_management" is unresponsive, so restarting the execution group.

Because message flow "Account_management" is unresponsive, the execution group is being restarted.

Review the message flow to see if there are any reasons why the flow has become unresponsive.

This looks familiar to the messages from the mqsistopmsgflow force command. It is informing us that the message flow is unresponsive and the execution group is restarting.

But notice that it is also telling us that this is due to a user configuration, the Workload Management properties for the message flow.



Level	Date and Time	Source
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:36	IBM Integration v9000
Warning	06/06/2013 16:27:35	IBM Integration v9000
Information	06/06/2013 16:27:33	IBM Integration v9000
Information	06/06/2013 16:27:33	IBM Integration v9000
Information	06/06/2013 16:24:17	IBM Integration v9000
Information	06/06/2013 16:21:57	IBM Integration v9000

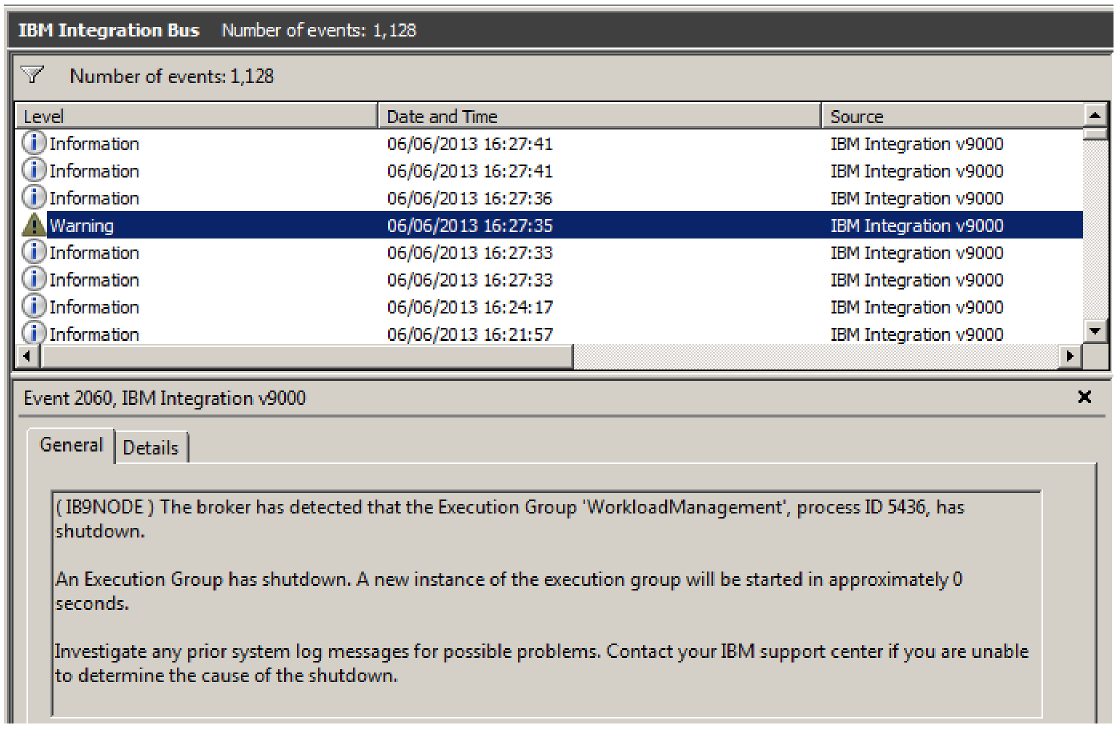
Event 7985, IBM Integration v9000

(IB9NODE.WorkloadManagement) Execution group "WorkloadManagement" is being restarted automatically due to a user configured action.

The execution group "WorkloadManagement" is being restarted in a controlled manner because of a condition set up by the user. See previous messages for the exact reason.

The execution group is being restarted because of a user action and does not require any additional actions to be taken.

9. Continue with the next message which is a warning message informing us that the execution group has shutdown. This was due to the property Processing action which was set to “Restart the execution group”.



Level	Date and Time	Source
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:36	IBM Integration v9000
Warning	06/06/2013 16:27:35	IBM Integration v9000
Information	06/06/2013 16:27:33	IBM Integration v9000
Information	06/06/2013 16:27:33	IBM Integration v9000
Information	06/06/2013 16:24:17	IBM Integration v9000
Information	06/06/2013 16:21:57	IBM Integration v9000

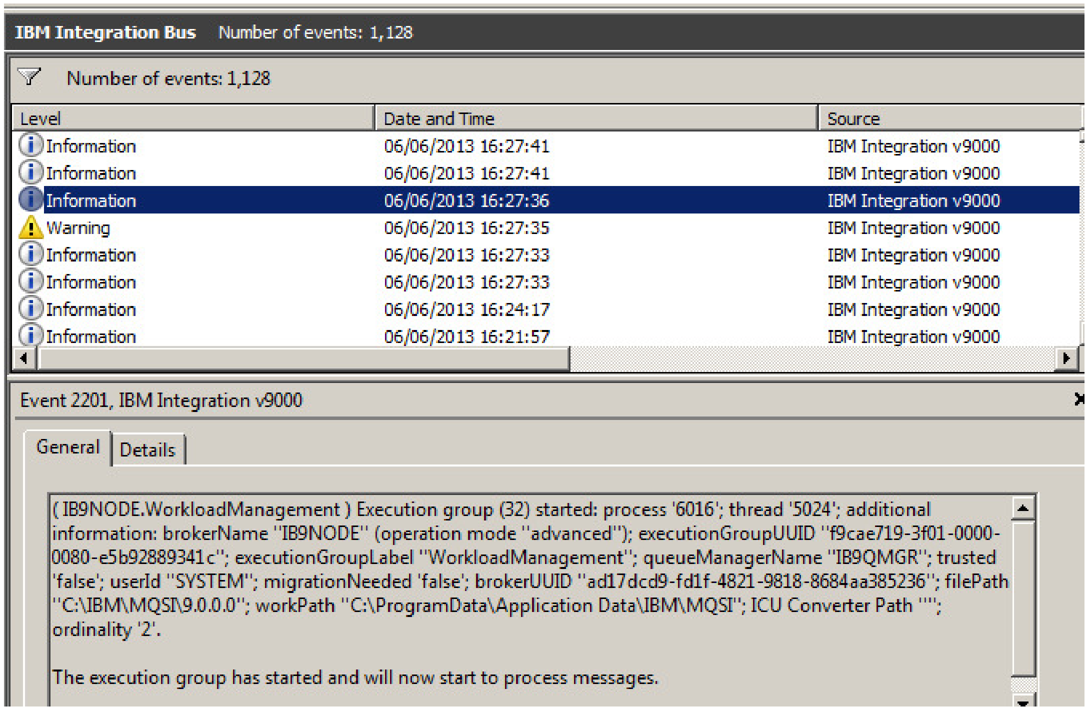
Event 2060, IBM Integration v9000

(IB9NODE) The broker has detected that the Execution Group 'WorkloadManagement', process ID 5436, has shutdown.

An Execution Group has shutdown. A new instance of the execution group will be started in approximately 0 seconds.

Investigate any prior system log messages for possible problems. Contact your IBM support center if you are unable to determine the cause of the shutdown.

10. The next message tells us the execution group has started.



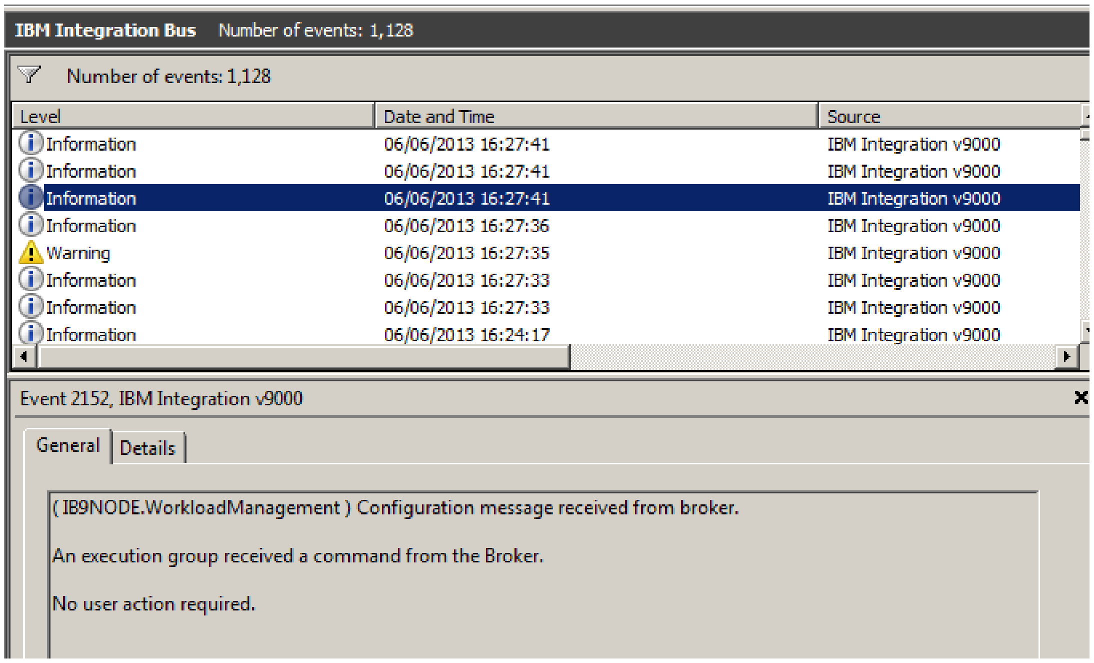
Level	Date and Time	Source
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:36	IBM Integration v9000
Warning	06/06/2013 16:27:35	IBM Integration v9000
Information	06/06/2013 16:27:33	IBM Integration v9000
Information	06/06/2013 16:27:33	IBM Integration v9000
Information	06/06/2013 16:24:17	IBM Integration v9000
Information	06/06/2013 16:21:57	IBM Integration v9000

Event 2201, IBM Integration v9000

(IB9NODE.WorkloadManagement) Execution group (32) started: process '6016'; thread '5024'; additional information: brokerName ''IB9NODE'' (operation mode ''advanced''); executionGroupUUID ''f9cae719-3f01-0000-0080-e5b92889341c''; executionGroupLabel ''WorkloadManagement''; queueManagerName ''IB9QMGR''; trusted 'false'; userId ''SYSTEM''; migrationNeeded 'false'; brokerUUID ''ad17dcd9-fd1f-4821-9818-8684aa385236''; filePath ''C:\IBM\MQSI\9.0.0.0''; workPath ''C:\ProgramData\Application Data\IBM\MQSI''; ICU Converter Path ''''; ordinality '2'.

The execution group has started and will now start to process messages.

11. The next three messages are concerning the restart of the execution group and configuration messages.



IBM Integration Bus Number of events: 1,128

Number of events: 1,128

Level	Date and Time	Source
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:36	IBM Integration v9000
Warning	06/06/2013 16:27:35	IBM Integration v9000
Information	06/06/2013 16:27:33	IBM Integration v9000
Information	06/06/2013 16:27:33	IBM Integration v9000
Information	06/06/2013 16:24:17	IBM Integration v9000

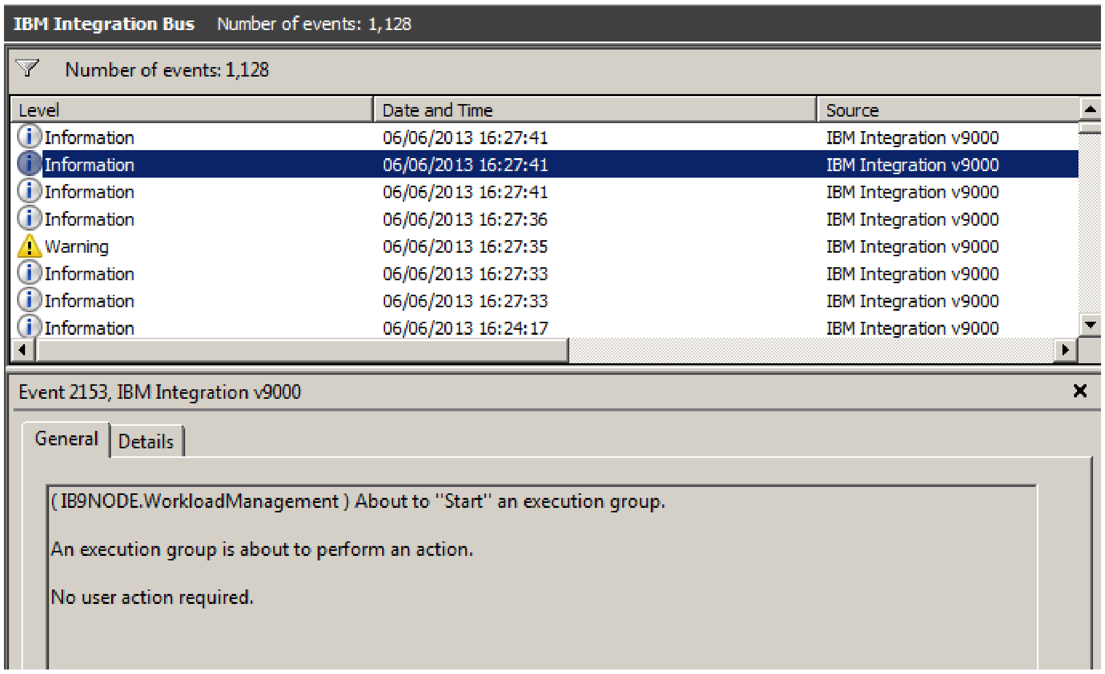
Event 2152, IBM Integration v9000

General Details

(IB9NODE.WorkloadManagement) Configuration message received from broker.

An execution group received a command from the Broker.

No user action required.



IBM Integration Bus Number of events: 1,128

Number of events: 1,128

Level	Date and Time	Source
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:36	IBM Integration v9000
Warning	06/06/2013 16:27:35	IBM Integration v9000
Information	06/06/2013 16:27:33	IBM Integration v9000
Information	06/06/2013 16:27:33	IBM Integration v9000
Information	06/06/2013 16:24:17	IBM Integration v9000

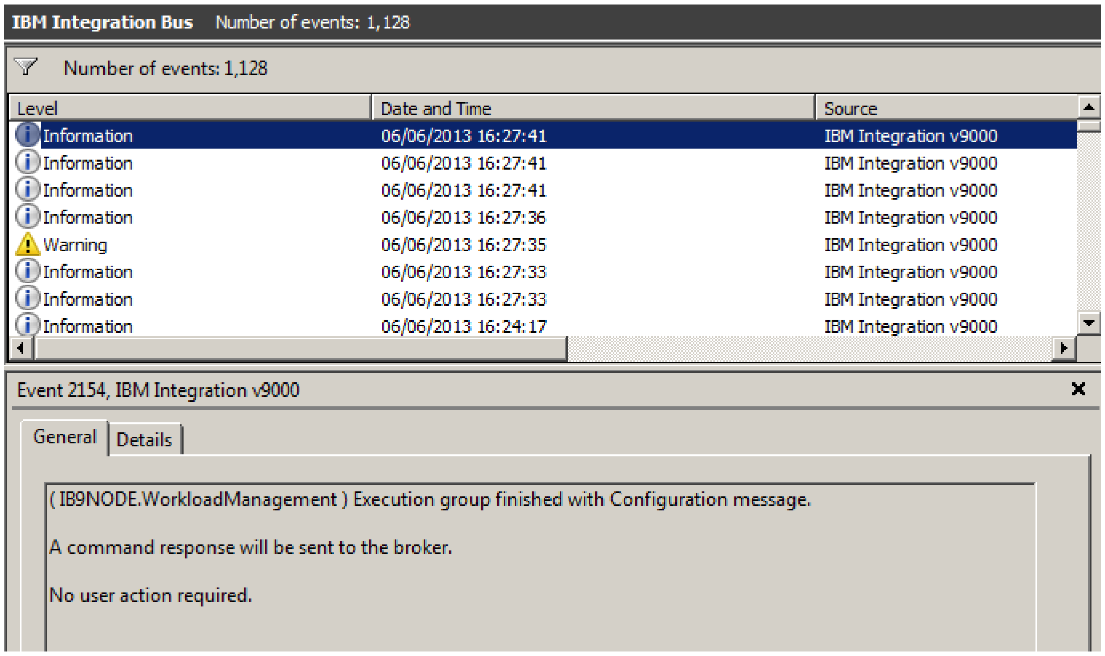
Event 2153, IBM Integration v9000

General Details

(IB9NODE.WorkloadManagement) About to "Start" an execution group.

An execution group is about to perform an action.

No user action required.



IBM Integration Bus Number of events: 1,128

Number of events: 1,128

Level	Date and Time	Source
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:41	IBM Integration v9000
Information	06/06/2013 16:27:36	IBM Integration v9000
Warning	06/06/2013 16:27:35	IBM Integration v9000
Information	06/06/2013 16:27:33	IBM Integration v9000
Information	06/06/2013 16:27:33	IBM Integration v9000
Information	06/06/2013 16:24:17	IBM Integration v9000

Event 2154, IBM Integration v9000

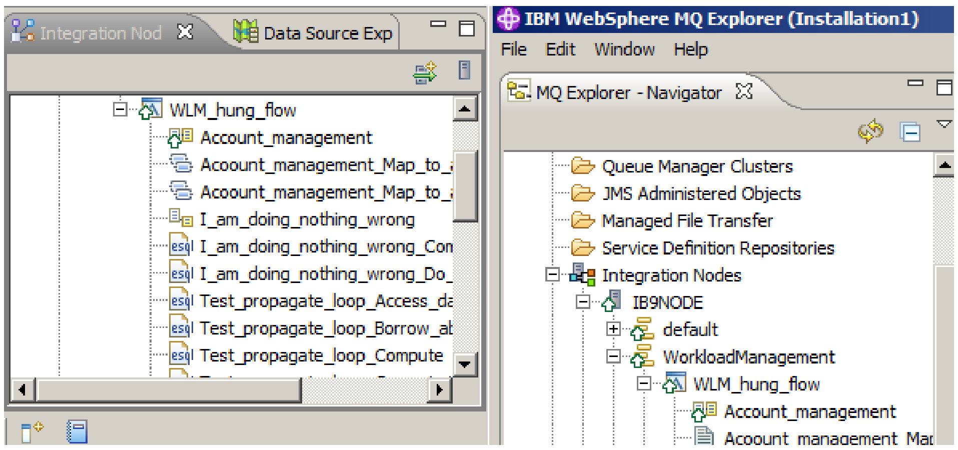
General | Details

(IB9NODE.WorkloadManagement) Execution group finished with Configuration message.

A command response will be sent to the broker.

No user action required.

12. Using the properties of the flow to automatically stop a message flow results in different behavior. Return the IIB Explorer in MQ Explorer or the IIB V9 Toolkit and observe the status of the integration node, 'WorkloadManagement' execution group, WLM_hung_flow application, and Account_management message flow. You will see that all are active.



Are you surprised by this behavior? When automatically stopping a flow by property settings, we are telling the integration node that the flow should be stopped after a specified amount of processing time. And the execution group was restarted due the setting for processing action. In this case, we don't know if the message flow was in a loop, only that it used too many cycles, and should be stopped. But when the execution group was restarted, the application and flow were also restarted because that was their previous status. When restarted, it may again be automatically restarted or continue processing until finished. We still have the option to manually force it out completely.

5. Subscribing to notifications of time out processing

Integration Bus V9 publishes messages for various workload conditions. Two of those publications are “processingTimeout” alerts and “processingFinished” alerts for when the message flow processing time-out period is exceeded and when message flow processing has completed when no action for timeout is specified.

The topic strings are available for subscriptions to receive the alerts. You can define subscriptions with the MQ Explorer or write your own applications to subscribe to the publications for the execution groups, applications, and message flows which you are interested in.

5.1 Message flow timeout exceeded event message

Below we describe the conditions and details for the event message that is published when the processingTimeoutSec timeout period is exceeded.

Once the processingTimeoutSec timeout period is exceeded for the message flow, a timeout exceeded XML event message is published.

You can view the message by subscribing to the following topic:

```
$SYS/Broker/<brokerName>/WorkloadManagement/ProcessingTimeout/<executionGroup
Label>/<applicationName>/<libraryName>/<messageFlowLabel>
```

where brokerName is the name of the broker, executionGroupLabel is the name of the execution group on that broker, applicationName is the name of the application on that execution group, libraryName is the name of the library on that application, and messageFlowLabel is the name of the message flow that is deployed to the library.

In the situation where the message flow is not contained in either an application or a library, the applicationName or libraryName parameters must be omitted along with their enclosing forward slash (/). For example:

If the message flow is not contained in an application and a library:

```
$SYS/Broker/<brokerName>/WorkloadManagement/ProcessingTimeout/<executio
nGroupLabel>/<messageFlowLabel>
```

If the message flow is contained in an application and not in a library:

```
$SYS/Broker/<brokerName>/WorkloadManagement/ProcessingTimeout/<executio
nGroupLabel>/<applicationName>/<messageFlowLabel>
```

Thereafter, if the processingTimeoutAction option has been set to none and processing of the message flow continues to completion, another event message is published that the processing has finished.

However, if the processingTimeoutAction option has been set to restartExecutionGroup the execution group is restarted and no further event messages are published from the message flow.

5.2 Message flow processing finished event message

Below we describe the conditions and details for the event message that is published if a timeout action of none is specified and when the message flow processing finishes.

If the message flow exceeds the *processingTimeoutSec* timeout period, the action that is defined by the property *processingTimeoutAction* is taken. If the action defined is none, processing of the message flow is allowed to continue and a processing finished XML event message is published once all processing has completed.

You can view the message by subscribing to the following topic:
$SYS/Broker/<brokerName>/WorkloadManagement/ProcessingFinished/<executionGroupLabel>/<applicationName>/<libraryName>/<messageFlowLabel>

where brokerName is the name of the broker, executionGroupLabel is the name of the execution group on that broker, applicationName is the name of the application on that execution group, libraryName is the name of the library on that application, and messageFlowLabel is the name of the message flow that is deployed to the library.

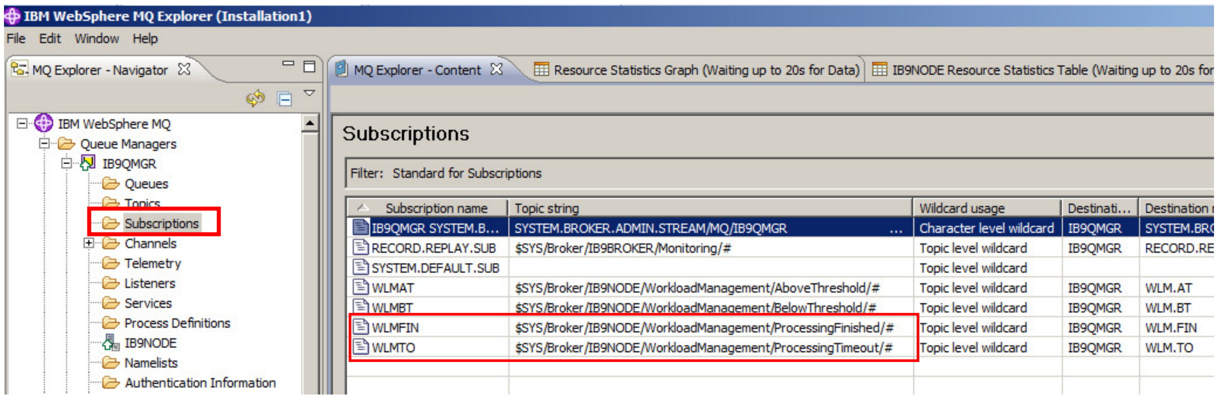
In the situation where the message flow is not contained in either an application or a library, the applicationName or libraryName parameters must be omitted along with their enclosing forward slash (/). For example:

If the message flow is not contained in an application and a library:
$SYS/Broker/<brokerName>/WorkloadManagement/ProcessingFinished/<executionGroupLabel>/<messageFlowLabel>

If the message flow is contained in an application and not in a library:
$SYS/Broker/<brokerName>/WorkloadManagement/ProcessingFinished/<executionGroupLabel>/<applicationName>/<messageFlowLabel>

5.3 Review the alerts

1. When the MQ resources were defined, two subscriptions were created for monitoring the notification threshold. The two subscriptions are WLMTO and WLMFIN. Return to MQ Explorer and click on Subscriptions.



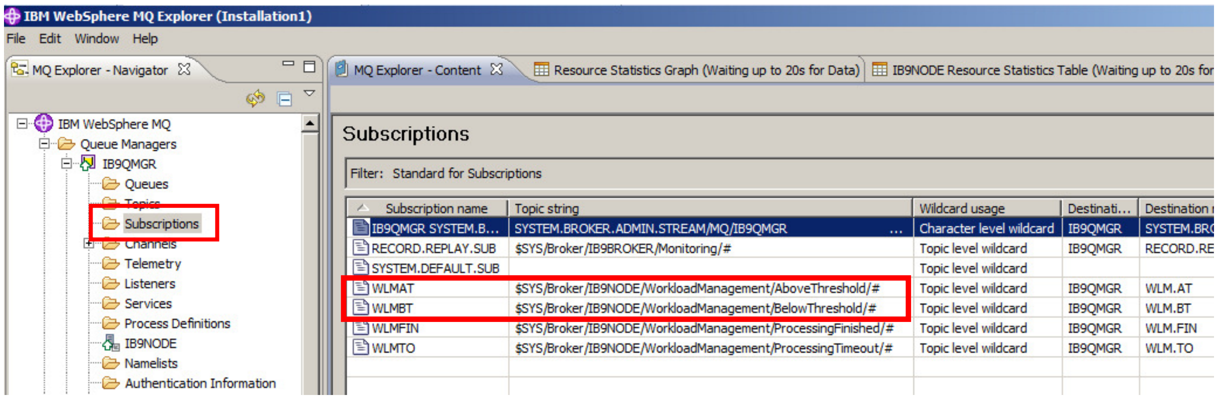
Subscriptions

Filter: Standard for Subscriptions

Subscription name	Topic string	Wildcard usage	Destinati...	Destination
IB9QMGR SYSTEM.B...	SYSTEM.BROKER.ADMIN.STREAM/MQ/IB9QMGR ...	Character level wildcard	IB9QMGR	SYSTEM.BR
RECORD.REPLAY.SUB	$SYS/Broker/IB9BROKER/Monitoring/#	Topic level wildcard	IB9QMGR	RECORD.RE
SYSTEM.DEFAULT.SUB		Topic level wildcard		
WLMAT	$SYS/Broker/IB9NODE/WorkloadManagement/AboveThreshold/#	Topic level wildcard	IB9QMGR	WLM.AT
WLMBT	$SYS/Broker/IB9NODE/WorkloadManagement/BelowThreshold/#	Topic level wildcard	IB9QMGR	WLM.BT
WLMFIN	$SYS/Broker/IB9NODE/WorkloadManagement/ProcessingFinished/#	Topic level wildcard	IB9QMGR	WLM.FIN
WLMTO	$SYS/Broker/IB9NODE/WorkloadManagement/ProcessingTimeout/#	Topic level wildcard	IB9QMGR	WLM.TO

2. You can see that the subscription WLMTO subscribed to alert publications for the topic string *$SYS/Broker/IB9NODE/WorkloadManagement/ProcessingTimeout/#* published by the broker. The hash is a wild card so it will receive "timeout" alerts for all execution groups, all applications, and all message flows which have been deployed with a ProcessingTimeout property set. Observe the queue names for the subscriptions WLMTO (WLM.TO) and WLMFIN (WLM.FIN).

   Note: You may need to scroll to the right to see the Destination Queue Manager and Destination columns. Some columns were hidden in the screen shot below.

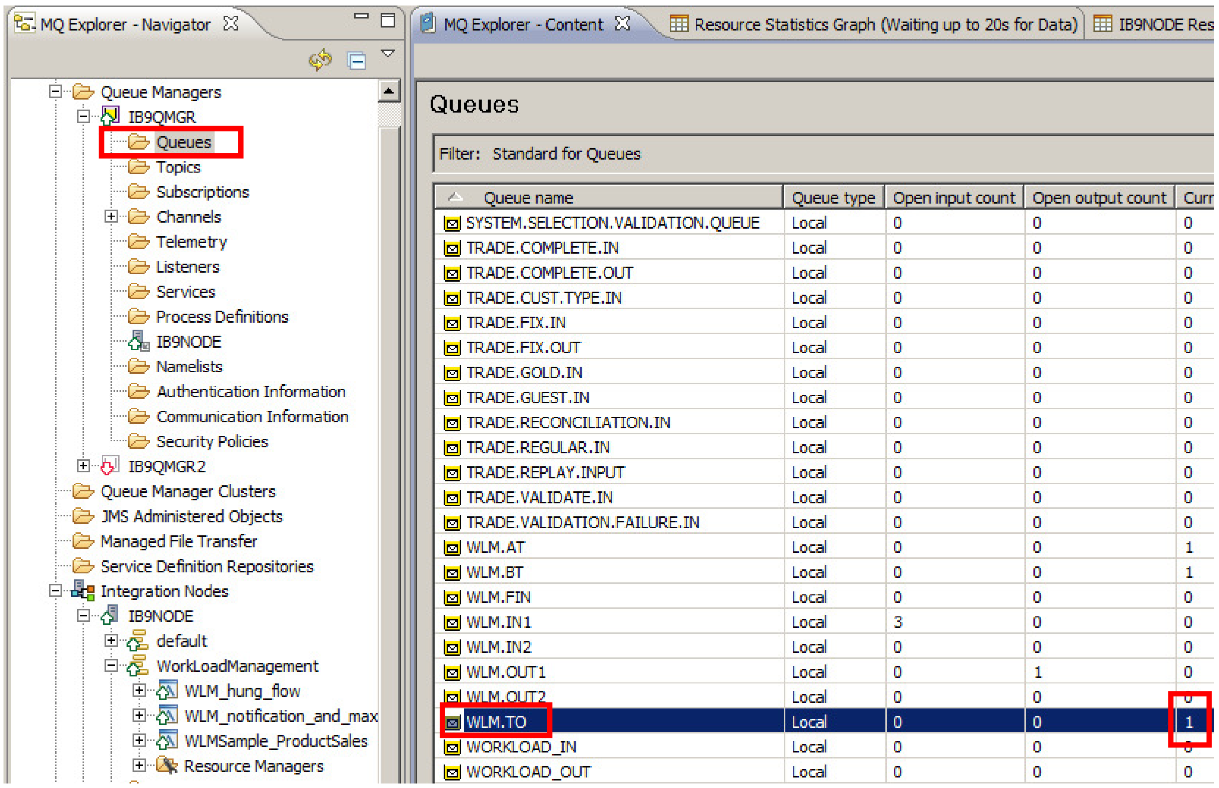


Subscriptions

Filter: Standard for Subscriptions

Subscription name	Topic string	Wildcard usage	Destinati...	Destination
IB9QMGR SYSTEM.B...	SYSTEM.BROKER.ADMIN.STREAM/MQ/IB9QMGR ...	Character level wildcard	IB9QMGR	SYSTEM.BRO
RECORD.REPLAY.SUB	$SYS/Broker/IB9BROKER/Monitoring/#	Topic level wildcard	IB9QMGR	RECORD.RE
SYSTEM.DEFAULT.SUB		Topic level wildcard		
WLMAT	$SYS/Broker/IB9NODE/WorkloadManagement/AboveThreshold/#	Topic level wildcard	IB9QMGR	WLM.AT
WLMBT	$SYS/Broker/IB9NODE/WorkloadManagement/BelowThreshold/#	Topic level wildcard	IB9QMGR	WLM.BT
WLMFIN	$SYS/Broker/IB9NODE/WorkloadManagement/ProcessingFinished/#	Topic level wildcard	IB9QMGR	WLM.FIN
WLMTO	$SYS/Broker/IB9NODE/WorkloadManagement/ProcessingTimeout/#	Topic level wildcard	IB9QMGR	WLM.TO

3. **Click on Queues and locate the WLM* queues.**



Queues

Filter: Standard for Queues

Queue name	Queue type	Open input count	Open output count	Curr
SYSTEM.SELECTION.VALIDATION.QUEUE	Local	0	0	0
TRADE.COMPLETE.IN	Local	0	0	0
TRADE.COMPLETE.OUT	Local	0	0	0
TRADE.CUST.TYPE.IN	Local	0	0	0
TRADE.FIX.IN	Local	0	0	0
TRADE.FIX.OUT	Local	0	0	0
TRADE.GOLD.IN	Local	0	0	0
TRADE.GUEST.IN	Local	0	0	0
TRADE.RECONCILIATION.IN	Local	0	0	0
TRADE.REGULAR.IN	Local	0	0	0
TRADE.REPLAY.INPUT	Local	0	0	0
TRADE.VALIDATE.IN	Local	0	0	0
TRADE.VALIDATION.FAILURE.IN	Local	0	0	0
WLM.AT	Local	0	0	1
WLM.BT	Local	0	0	1
WLM.FIN	Local	0	0	0
WLM.IN1	Local	3	0	0
WLM.IN2	Local	0	0	0
WLM.OUT1	Local	0	1	0
WLM.OUT2	Local	0	0	0
WLM.TO	Local	0	0	1
WORKLOAD_IN	Local	0	0	0
WORKLOAD_OUT	Local	0	0	0

4. Look to see if you have any messages in the WLM.TO and WLM.FIN. You should have received some during the above tests.

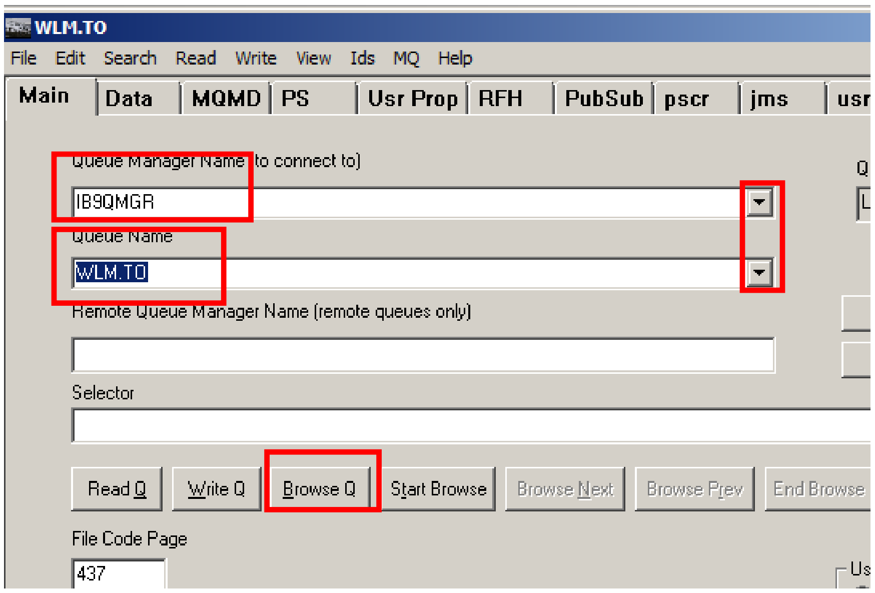
   During my test, I received one “processingTimeout” alert when the value for Processing Timeout was set to fifteen seconds. The flow was automatically removed from the system and the execution group was restarted. When the flow was stopped and the execution group restarted, the broker published the “processingTimeout” alert.

   At the current time restartExecutionGroup is the only available action for Processing Timeout Action. It cannot be set to None, so we cannot see the “processingFinished” message.

5. Open RHFUtil from the Start button.

   Use the pull-downs to populate the Queue Manager Name with IB9QMGR and the Queue Name with WLM.TO.

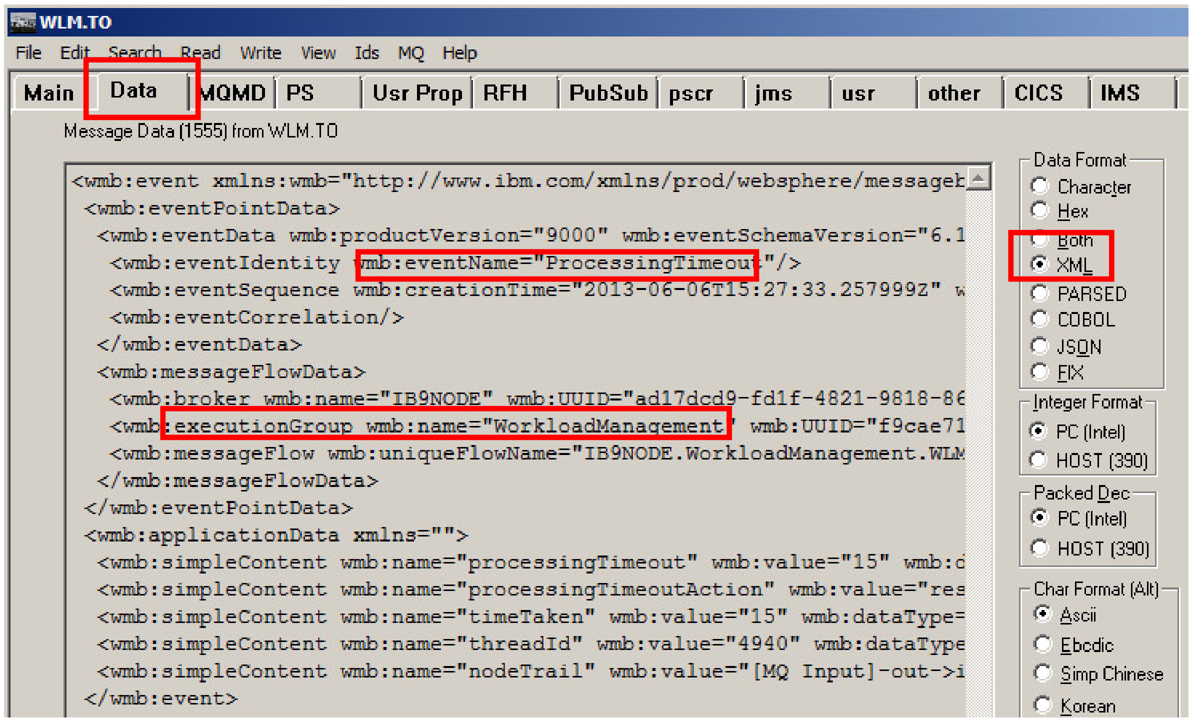
   Click Browse Q.


WLM.TO
File Edit Search Read Write View Ids MQ Help
Main Data MQMD PS Usr Prop RFH PubSub pscr jms usr
Read Q
Write Q
Browse Q
Start Browse
Browse Next
Browse Prev
End Browse

Queue Manager Name (to connect to): IB9QMGR
Queue Name: WLM.TO
Remote Queue Manager Name (remote queues only):
Selector:
File Code Page: 437

6. Click on the Data tab at the top of the window and click on XML under Data Format to format the data.

   Review the XML message which was received. The event was an “ProcessingTimeout”. The execution group is identified – WorkloadManagement.


WLM.TO
File Edit Search Read Write View Ids MQ Help
Main Data MQMD PS Usr Prop RFH PubSub pscr jms usr other CICS IMS
Data Format
Character
Hex
Both
Integer Format
Packed Dec
Char Format (Alt)

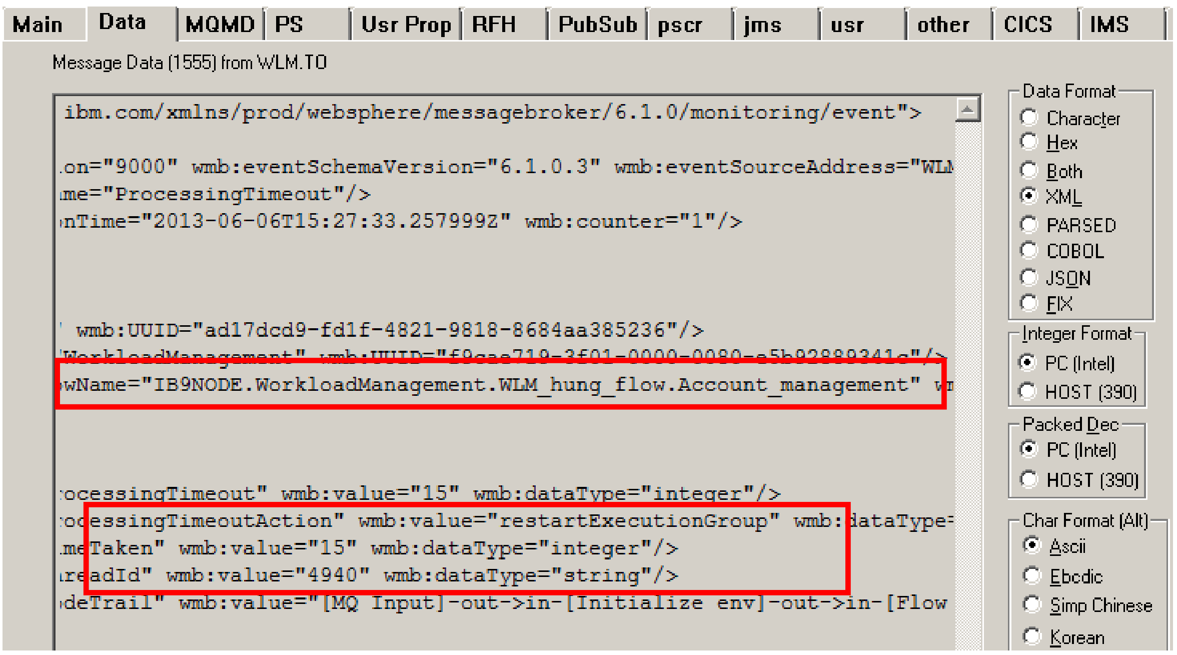
Message Data (1555) from WLM.TO

```
<wmb:event xmlns:wmb="http://www.ibm.com/xmlns/prod/websphere/messageb
 <wmb:eventPointData>
  <wmb:eventData wmb:productVersion="9000" wmb:eventSchemaVersion="6.1
   <wmb:eventIdentity wmb:eventName="ProcessingTimeout"/>
   <wmb:eventSequence wmb:creationTime="2013-06-06T15:27:33.257999Z" w
   <wmb:eventCorrelation/>
  </wmb:eventData>
  <wmb:messageFlowData>
   <wmb:broker wmb:name="IB9NODE" wmb:UUID="ad17dcd9-fd1f-4821-9818-86
   <wmb:executionGroup wmb:name="WorkloadManagement" wmb:UUID="f9cae71
   <wmb:messageFlow wmb:uniqueFlowName="IB9NODE.WorkloadManagement.WLM
  </wmb:messageFlowData>
 </wmb:eventPointData>
 <wmb:applicationData xmlns="">
  <wmb:simpleContent wmb:name="processingTimeout" wmb:value="15" wmb:d
  <wmb:simpleContent wmb:name="processingTimeoutAction" wmb:value="res
  <wmb:simpleContent wmb:name="timeTaken" wmb:value="15" wmb:dataType=
  <wmb:simpleContent wmb:name="threadId" wmb:value="4940" wmb:dataType
  <wmb:simpleContent wmb:name="nodeTrail" wmb:value="[MQ Input]-out->i
 </wmb:event>
```

Data Format: XML; Integer Format: PC (Intel); Packed Dec: PC (Intel); Char Format (Alt): Ascii

Options: Character, Hex, Both, XML, PARSED, COBOL, JSON, FIX; PC (Intel), HOST (390); Ascii, Ebcdic, Simp Chinese, Korean

You will need to scroll to the right to see all of the fields. After scrolling right, you will see the application and message flow names. You can also see the processingTimeout value for the message flow and the action restartExecutionGroup. It also reports the timetaken, threadID, and the nodeTrail. You will need to scroll further right to see more of the trail.


Main Data MQMD PS Usr Prop RFH PubSub pscr jms usr other CICS IMS
Data Format
Integer Format
Packed Dec
Char Format (Alt)

Message Data (1555) from WLM.TO

```
ibm.com/xmlns/prod/websphere/messagebroker/6.1.0/monitoring/event">

on="9000" wmb:eventSchemaVersion="6.1.0.3" wmb:eventSourceAddress="WLM
me="ProcessingTimeout"/>
nTime="2013-06-06T15:27:33.257999Z" wmb:counter="1"/>



' wmb:UUID="ad17dcd9-fd1f-4821-9818-8684aa385236"/>
WorkloadManagement" wmb:UUID="f9cae719-3f01-0000-0080-e5b92889341c"/>
wName="IB9NODE.WorkloadManagement.WLM_hung_flow.Account_management" wm



ocessingTimeout" wmb:value="15" wmb:dataType="integer"/>
ocessingTimeoutAction" wmb:value="restartExecutionGroup" wmb:dataType=
meTaken" wmb:value="15" wmb:dataType="integer"/>
readId" wmb:value="4940" wmb:dataType="string"/>
deTrail" wmb:value="[MQ Input]-out->in-[Initialize env]-out->in-[Flow
```

Data Format: XML; Integer Format: PC (Intel); Packed Dec: PC (Intel); Char Format (Alt): Ascii

6. Unresponsive Flows Summary

In this lab, you saw that there are now three possible methods to stop a runaway message flow and remove it from the execution group.

1. Programmatically check from within a message flow to see if it has been requested to stop
2. Manually force a message flow to stop
3. Automatically force a message flow to stop

We briefly introduced using the APIs to see if a flow has been requested to stop. We then practiced the other two methods and observed the behavior of the integration node.